



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

NÁVRH ELEKTRONICKÉHO SUBSYSTÉMU PRO SIMULÁTOR DOPADU VE SNÍŽENÉ GRAVITACI

DESIGN OF ELECTRONIC SUBSYSTEM FOR REDUCED GRAVITY IMPACT SIMULATOR

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. LUBOMÍR OSTRÝ

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. STANISLAV VĚCHET, Ph.D.

BRNO 2020

Zadání diplomové práce

Ústav: Ústav mechaniky těles, mechatroniky a biomechaniky
Student: **Bc. Lubomír Ostrý**
Studijní program: Aplikované vědy v inženýrství
Studijní obor: Mechatronika
Vedoucí práce: **doc. Ing. Stanislav Věchet, Ph.D.**
Akademický rok: 2019/20

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Návrh elektronického subsystému pro simulátor dopadu ve snížené gravitaci

Stručná charakteristika problematiky úkolu:

Přistávací manévry na cizím kosmickém tělese je značně komplikovaný proces, jak bylo doloženo v případě neúspěšného přistání landeru Philae. Pro maximalizaci spolehlivosti přistání je důležité provést sérii testů navržených přistávacích modulů. Za tímto účelem se běžně používají složitá mechanická zařízení založená na různých principech, každé z nich však skrývá jisté nevýhody. Cílem práce je návrh elektronického subsystému pro simulátor dopadu ve snížené gravitaci, který bude mít za úkol řízení a sběr dat během experimentu.

Cíle diplomové práce:

- 1) Rešerše možností řízení a sběru dat v podobných aplikacích.
- 2) Návrh elektronického subsystému pro řízení simulátoru dopadu.
- 3) Řízení experimentu, měření vybraných veličin.
- 4) Vytvoření uživatelského grafického rozhraní.
- 5) Vyhodnocení a prezentace výsledků.

Seznam doporučené literatury:

BISHOP, Robert H. Mechatronic System Control, Logic, and Data Acquisition Boca Raton, FL: CRC Press, c2008. ISBN 9780849392603.

ZHANG, Peng. Advanced Industrial Control Technology. Binghamton: Elsevier Science & Technology Books, 2010. DOI: 10.1016/C2009-0-20337-0. ISBN 9781437778076.

NORRIS, Donald. Programming with STM32: getting started with the Nucleo Board and C/C++. New York: McGraw-Hill Education, [2018]. ISBN 1260031314.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2019/20

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt

Tato práce se zaměřuje na vývoj elektronického subsystému pro zařízení simulující dopad při podmínkách snížené gravitace. V úvodu se řešeršní část práce nejprve zabývá průzkumem možností simulace mikrogravitace a snížené gravitace a jejich porovnáním, v další části jsou charakterizovány a porovnány tři vybrané potenciální přístupy pro vytvoření elektronického subsystému simulátoru dopadu. V praktické části této diplomové práce je popsán návrh a realizace zmíněného elektronického subsystému. Základem elektronického subsystému je řídicí jednotka, která byla vytvořena na základě mikrokontroleru od společnosti STMicroelectronics, pomocí ní je v systému snímán tlak, poloha, zrychlení, síla a dále je řídicí jednotka využita pro řízení krokového motoru. U každého prvku je popsána jeho integrace do systému jak z hlediska hardwaru, tak i softwaru. Pro interakci se systémem bylo dále vyvinuto grafické uživatelské rozhraní pro PC, na které se zaměřuje další z kapitol. V závěru práce je popsána činnost elektronického subsystému v průběhu experimentu a následně je navržený subsystém zhodnocen.

Summary

The focus of this thesis is development of an electronic subsystem for reduced gravity impact simulator. The research part of this thesis firstly covers methods used for simulation of microgravity or reduced gravity and compares them. Another part of research focuses on three selected potential approaches to creating the electronic subsystem for this device. The second, practical, part of this thesis describes the design and development of the electronic subsystem. The foundation of the electronic subsystem is a control unit which has been developed on the basis of an STMicroelectronics microcontroller. Using the control unit, the electronic subsystem measures pressure, position, acceleration and force in the system. Another task of the control unit is control of a stepper motor. Integration of individual elements into the electronic subsystem is described both in terms of software and hardware. Furthermore, a graphic user interface program for PC has been developed as a means to interact with the system. In the final part of the thesis, the operation of the electronic subsystem is described and lastly, the electronic subsystem is evaluated.

Klíčová slova

Řídicí systém, akvizice dat, STM32, kompenzace gravitační síly, snížená gravitace, mikrogravitace

Keywords

Control system, data acquisition, STM32, gravity offload, reduced gravity, microgravity

OSTRÝ, L. *Návrh elektronického subsystému pro simulátor dopadu ve snížené gravitaci*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2020. 63 s. Vedoucí doc. Ing. Stanislav Věchet, Ph.D.

Prohlašuji, že jsem diplomovou práci *Návrh elektronického subsystému pro simulátor dopadu ve snížené gravitaci* vypracoval samostatně pod vedením doc. Ing. Stanislava Věcheta, Ph.D., s použitím materiálů uvedených v seznamu literatury.

Bc. Lubomír Ostrý

Děkuji společnosti Sobriety s.r.o., jejím zaměstnancům a mému vedoucímu doc. Ing. Stanislavu Věchetovi, Ph.D. za podporu a odborné vedení mé diplomové práce.

Bc. Lubomír Ostrý

Obsah

1	Úvod	3
2	Způsoby simulace mikrogravitace	5
2.1	Parabolické lety	5
2.2	Experimenty na oběžné dráze	5
2.3	Padací věže	6
2.4	Sondážní rakety	7
2.5	Systémy kompenzace tíhové síly	7
2.5.1	Vztlakové metody	8
2.5.2	Mechanické metody	8
3	Projekt Zero-G	12
3.1	Princip	12
3.2	Požadavky na elektronický subsystém	14
4	Možnosti řízení a sběru dat v podobných aplikacích	17
4.1	PLC	17
4.2	Vestavěné systémy	19
4.3	Měření a řízení pomocí PC	21
5	Návrh elektronického subsystému	23
5.1	Řídicí jednotka	23
5.1.1	STM32F767ZI	24
5.2	Vývojové prostředí	24
5.3	Obecná struktura C programu	25
5.4	Snímání tlaku	26
5.4.1	Charakteristika snímačů tlaku	26
5.4.2	EIA/TIA-485	26
5.4.3	Hardwarová integrace	27
5.4.4	MODBUS protokol	28
5.4.5	Softwarová integrace	29
5.5	Snímání polohy	31
5.5.1	Charakteristika snímačů polohy	32
5.5.2	Integrace snímání polohy	32
5.5.3	Odhad rychlosti	33
5.6	Snímání zrychlení	34
5.6.1	Snímač	34
5.6.2	I2C	35
5.6.3	Integrace snímače zrychlení	35
5.7	Snímání síly	36
5.7.1	Hardware	36
5.7.2	SPI komunikace	37
5.7.3	Softwarová integrace	38
5.8	Ovládání motorů	38
5.8.1	Hardware	38

5.8.2	Softwarová integrace	39
5.9	Ovládání solenoidové západky	40
5.10	Komunikace s PC	41
5.11	Napájení systému	42
5.12	Realizace hardwaru	43
6	Návrh grafického uživatelského rozhraní	44
6.1	Qt	45
6.2	Struktura programu uživatelského rozhraní	45
6.2.1	Sériová komunikace	46
6.2.2	Grafické rozhraní	46
6.3	Grafický návrh a funkcionality	47
7	Řízení a průběh experimentu	50
8	Zhodnocení	51
9	Závěr	53

1. Úvod

Se začátkem dobývání vesmíru před více než půl stoletím současně vznikla i potřeba pro testování zařízení a jejich součástí, které do kosmu vysíláme. Testování z důvodu náročných podmínek, často značně odlišných od těch na Zemi, kterým budou tato zařízení vystavena. Jednou z podmínek, kterou je zvláště náročné napodobit pro potřeby testování, je gravitace, respektive gravitační zrychlení rozdílné od toho Zemského a dále v extrémním případě, napodobení absence gravitace – mikrogravitace.

Termín mikrogravitace je běžně používán jako synonymum ke slovnímu spojení stav beztlíže nebo nulová gravitace, pojem mikrogravitace však označuje stav, kdy je těleso „téměř“ ve stavu nulové gravitace, ale stále na něj působí relativně malé zbytkové gravitační zrychlení, řádově miliontiny gravitačního zrychlení na Zemi. Mikrogravitace se také někdy používá jako jednotka a udává kolik miliontin Zemského gravitačního zrychlení působí v daném prostředí.[1][2]

Pro dosažení podmínek mikrogravitace bylo vymyšleno mnoho zařízení a způsobů využívající různých principů, každé řešení má však řadu výhod a nevýhod. Jednou z možností pro provádění experimentů v podmínkách mikrogravitace je umístění experimentu na oběžnou dráhu, jako například experimenty prováděné na Mezinárodní vesmírné stanici. Dalšími možnostmi jsou například parabolické lety, využití sondážních raket, nebo tzv. drop towers neboli padací věže. Tyto metody využívají volného pádu pro vytvoření prostředí mikrogravitace, což s sebou ze své podstaty přináší mnoho překážek. Existují další způsoby simulace mikrogravitace, které spoléhají na kompenzaci tíhové síly, nadlehčováním daného tělesa. Úvodní kapitola pojednává o používaných způsobech vytváření mikrogravitace více podrobně a nabízí jejich srovnání.[3][4][5][6]

Tato práce se podílí na projektu s pracovním označením Zero-G společnosti Sobriety s.r.o., který se zaměřuje na vývoj zařízení pro experimentální testování dopadů za podmínek simulované snížené gravitace a mikrogravitace. Toto testovací zařízení slouží jako tzv. "proof of concept", jeho účelem je tedy ověření využitelnosti použitého principu, z tohoto důvodu je zařízení navrhováno ve zmenšeném měřítku. V případě úspěchu zmenšené verze, by v budoucnu mohlo zařízení v plném měřítku provádět dopadové testování při simulované mikrogravitaci a snížené gravitaci na rozměrných a hmotných objektech. Příkladem takovýchto experimentů by mohlo být testování přistávacích modulů v plném měřítku, tento přístup by umožnil přesněji předpovídat důsledky dopadu v daných podmínkách a prověření přistávacích systémů.

Příkladem motivace pro vytvoření projektu, na kterém se podílí tato diplomová práce, byly komplikace při přistání sondy Philae během mise Rosetta Evropské kosmické agentury. Cílem této mise byl průzkum komety 67P/Čurjumov-Gerasimenko pomocí sondy Philae, která měla na povrchu komety přistát a provést řadu testů, jako například určení chemického a mineralogického složení povrchu komety. Během přistání sondy Philae však nastalo selhání přistávacích systémů a z důvodu velmi nízké gravitace komety došlo k odrazu sondy od povrchu.[7][8] S použitím systému pro simulaci dopadu ve snížené gravitaci

by bylo možné tuto a jiné dopadové situace simulovat a snížit tak riziko selhání během přistání.

Tato diplomová práce má za cíl návrh elektronického subsystému pro zmíněný testovací systém, který bude mít za úkol měření vybraných veličin během experimentu a také řízení samotného experimentu.

2. Způsoby simulace mikrogravitace

2.1. Parabolické lety

Parabolické lety, jak název napovídá, jsou způsobem simulace mikrogravitace pomocí letu po parabolické křivce, s využitím upravených dopravních letadel. Při těchto letech dopravní letadlo nejprve během vstupní fáze stoupá s přetížením 1,8g a dále přechází do parabolické části, během které se ocitá na 22 sekund ve stavu mikrogravitace. Piloti mají na starosti přesné kopírování stanovené křivky, což je dosaženo upravováním tahu motorů, zásahy křidélek a stabilizátoru letadla. Po 22 sekundách mikrogravitace následuje výstupní fáze, během které letadlo vyrovnává let a je vystaveno přetížení 1,8g.[4]

Oproti jiným způsobům je simulace mikrogravitace pomocí parabolických letů méně kvalitní při úrovni mikrogravitace pohybující se běžně kolem 10^{-2} g. Další nevýhodou tohoto přístupu je velmi krátká doba, po kterou je možné mikrogravitaci simulovat – pouze 22 sekund. Během jednoho letu je avšak provedeno až několik desítek opakování a je tak možné nasbírat více než 10 minut dat testování v prostředí mikrogravitace.[9]

Parabolické lety jsou využívány pro experimenty, které nevyžadují dlouhé intervaly mikrogravitace nebo také pro prověření experimentů před jejich vysláním na dlouhodobé testování na oběžné dráze. Tento přístup je také používán pro testování vlivů mikrogravitace na lidský organismus.[9]

Výhodou simulace mikrogravitace parabolickými lety jsou relativně nízké náklady oproti ostatním řešením, což činí testování za podmínek mikrogravitace více dostupné. Modifikované dopravní letouny nabízí prostor pro provádění až přibližně 15 experimentů zároveň. Další výhodou je možnost zásahu do experimentu v jeho průběhu samotnými výzkumníky za účelem úprav nebo změn v nastavení. Oproti pokusům na oběžné dráze umožňuje prostředí dopravního letounu také díky méně přísným technickým omezením provádět testování s běžnými laboratorními přístroji nebo s rozměrným vybavením.[9]

Kromě simulace mikrogravitace je změnou trajektorie možné simulovat i jiné hodnoty gravitačního zrychlení, což činí z parabolických letů univerzální způsob testování v podmínkách snížené gravitace.[9]

2.2. Experimenty na oběžné dráze

Testování ve stavu mikrogravitace lze dosáhnout také prováděním experimentů na oběžné dráze, běžně například v prostředí Mezinárodní vesmírné stanice. Ve své podstatě tento přístup využívá stejného principu jako parabolické lety, padací věže nebo sondážní rakety – volného pádu. Objekty na oběžné dráze jsou vystaveny pouze gravitačnímu zrychlení Země, jsou tedy ve stavu volného pádu, avšak jejich tečná rychlost zajišťuje konstantní

2.3. PADACÍ VĚŽE

vzdálenost od Země. Principiálně je tedy možné na tuto metodu nahlížet jako na „trvalý volný pád“.[10]

Z tohoto plyne jedna z hlavních výhod provádění experimentů s mikrogravitací na oběžné dráze – možnost dlouhotrvajících experimentů s prakticky neomezenou dobou trvání. Další výhodou oproti ostatním metodám je úroveň mikrogravitace, pohybující se v oblasti 10^{-4} g v případě Mezinárodní vesmírné stanice. Odchylky od nulové gravitace zde jsou způsobeny zejména dalšími přístroji a pohybem posádky na Mezinárodní vesmírné stanici. Specializovaným výzkumným satelitem bez posádky lze dále dosáhnout ještě lepších úrovní mikrogravitace, řádově až 10^{-6} g. Tento přístup však ze své podstaty, na rozdíl od Mezinárodní vesmírné stanice (nebo jiných satelitů s posádkou), nedovoluje zasahování do experimentu výzkumníky. Z důvodu nepřetržitého stavu mikrogravitace a přítomnosti posádky na Mezinárodní vesmírné stanici je tento způsob ideální pro sledování dlouhodobých účinků mikrogravitace na lidský organizmus.[9][11]

Nevýhodou tohoto přístupu je velmi vysoká cena vyslání experimentálního vybavení na oběžnou dráhu a dále také výrazné objemové a hmotnostní omezení, vyplývající z prostředí satelitů a způsobu dopravy na oběžnou dráhu. Dalším omezením pro provádění experimentů na Mezinárodní vesmírné stanici jsou také striktní technické požadavky pro vyslané testovací zařízení.[9]

2.3. Padací věže

Podobně jako jiné metody, využívají i padací věže volného pádu k vytvoření mikrogravitace. Z důvodu eliminace odporu vzduchu, který by negativně ovlivňoval volný pád, jsou padací věže uvnitř tvořeny vakuovou komorou, ze které je během testování vyčerpán vzduch. Kapsle obsahující experiment je pak puštěna volným pádem z vrcholu věže a ve spodní části bezpečně zbrzděna a zachycena. Samotná délka trvání stavu mikrogravitace je pak z podstaty věci definována výškou dané věže. Čas mikrogravitace je možné prodloužit v případě, kdy je věž vybavena katapultem, který umožní kapsli vymrstit ze spodní části věže směrem vzhůru na začátku experimentu. Při tomto způsobu testování je kapsli udělena ve spodní části věže rychlost odpovídající dopadové rychlosti, které dosáhne kapsle při volném pádu z horní části věže. Kapsle tak navíc provádí pohyb inverzní k pohybu při pouhém vypuštění z horní části, výsledkem je zdvojnásobení trvání stavu mikrogravitace. Příkladem je padací věž Univerzity v Brémách, která měří celkově 145 m (s užitečnou výškou 110 m) a dosahuje trvání mikrogravitace 4,74 s. Při použití katapultu se zdvojnásobí na přibližně 9 s.[6]

Jednou z hlavních výhod tohoto typu testování v prostředí mikrogravitace je nízká cena. Nízká cena z důvodu možnosti provádění několika experimentů denně během celého roku. Padací věže jsou tak jedním z nejdostupnějších způsobů pro testování v prostředí mikrogravitace pro výzkumníky nebo v některých případech i pro studenty. Mimo nízké ceny nabízí padací věže také jednu z nejlepších úrovní mikrogravitace, která dosahuje až 10^{-6} g. [9][12]

Nevýhodou tohoto přístupu je velice krátká doba trvání mikrogravitace, pohybující se v jednotkách sekund. Další nevýhodou jsou objemové a hmotnostní omezení, které se odvíjí od konkrétní věže a kapsle.[9]

2.4. Sondážní rakety

Sondážní rakety jsou suborbitální rakety určené k provádění experimentů v nadmořských výškách přibližně 40 – 120 km. Prováděné vědecké experimenty se zabývají například pozorováním astronomických objektů nebo zkoumáním vrstev atmosféry. Díky volnému pádu, ve kterém se sondážní rakety ocitají po vypnutí raketových motorů, jsou tyto rakety vystaveny stavu mikrogravitace trvajícimu (v závislosti na dosažené nadmořské výšce) 5 – 20 minut, vytváří tak dále možnost provádění pokusů ve stavu mikrogravitace.[5]

Sondážní rakety dosahují mikrogravitace s přesností až 10^{-5} g, v úrovni mikrogravitace tak předčí například parabolické lety nebo Mezinárodní vesmírnou stanici. Oproti padacím věžím a parabolickým letům se jedná o dražší metodu testování, avšak stále méně nákladnou než vysílání experimentů na oběžnou dráhu. Sondážní rakety se proto často používají pro účely testování přístrojů před jejich vysláním na oběžnou dráhu.[11]

V závislosti na velikosti a typu rakety je možné vynést testovací zařízení vážící jednotky kilogramů až po zařízení o hmotnosti 500 kg. Rozměrově je experimentální zařízení omezeno rozměry samotné sondážní rakety. [5]

2.5. Systémy kompenzace tíhové síly

Systémy kompenzace tíhové síly je možné považovat za další způsob simulace mikrogravitace na Zemi, oproti ostatním popsaným způsobům vytváření prostředí mikrogravitace se však tento způsob zásadně liší použitým principem – na rozdíl od již zmíněných metod nevyužívají tyto systémy princip volného pádu. Systémy kompenzace tíhové síly spoléhají na vyvíjení kompenzační síly na zkoumané těleso, která má za účel nadlehčení tělesa a vyrušení, případně snížení gravitační síly. Tímto způsobem tak lze pro dané nadlehčované těleso simulovat prostředí mikrogravitace (nebo snížené gravitace v případě nižší hodnoty kompenzační síly), ze své podstaty však není možné tuto metodu aplikovat například na experimenty s kapalinami nebo plyny (jelikož síla má bodové, případně plošné působíště, nikoli objemové). Z tohoto důvodu je zřejmé, že tento způsob simulace mikrogravitace pomocí kompenzace tíhové síly je značně omezen ve svém použití, nabízí však oproti již zmíněným metodám určité výhody v závislosti na konkrétním provedení.[13]

2.5. SYSTÉMY KOMPENZACE TÍHOVÉ SÍLY

2.5.1. Vztlakové metody

Vztlakové metody jsou založeny na vytváření kompenzační síly pomocí vztlaku, který působí na těleso ponořené v kapalině.[13]

Jedním ze způsobů je ponoření testovaného tělesa nebo těles do vody, při tomto přístupu působí kompenzační síla (vztlaková síla) na těleso plošně. Výhodou této metody je její jednoduchost a možnost kompenzace tíhové síly objektů s velkou hmotností. Metoda ponoření tělesa do vody má na druhou stranu ze své podstaty mnoho nevýhod, kterými jsou například nutnost voděodolnosti testovaných zařízení a dále zejména vysoký odpor kapaliny při dynamickém testování. Z těchto důvodů není tato metoda běžně využívána pro testování samotných zařízení, ale převážně pouze pro výcvik astronautů.[15]

Jako vztlakovou metodu lze označit i metodu využívající heliové balony pro nadlehčení testovaného zařízení. Jelikož většinu vztlaku poskytují balony připevněné pomocí lan, je kompenzační síla působící na těleso pouze bodová. Nevýhodou této metody je také nízká únosnost heliových balonů a velkých odpor vzduchu působící na heliové balony při pohybu. [14][15]

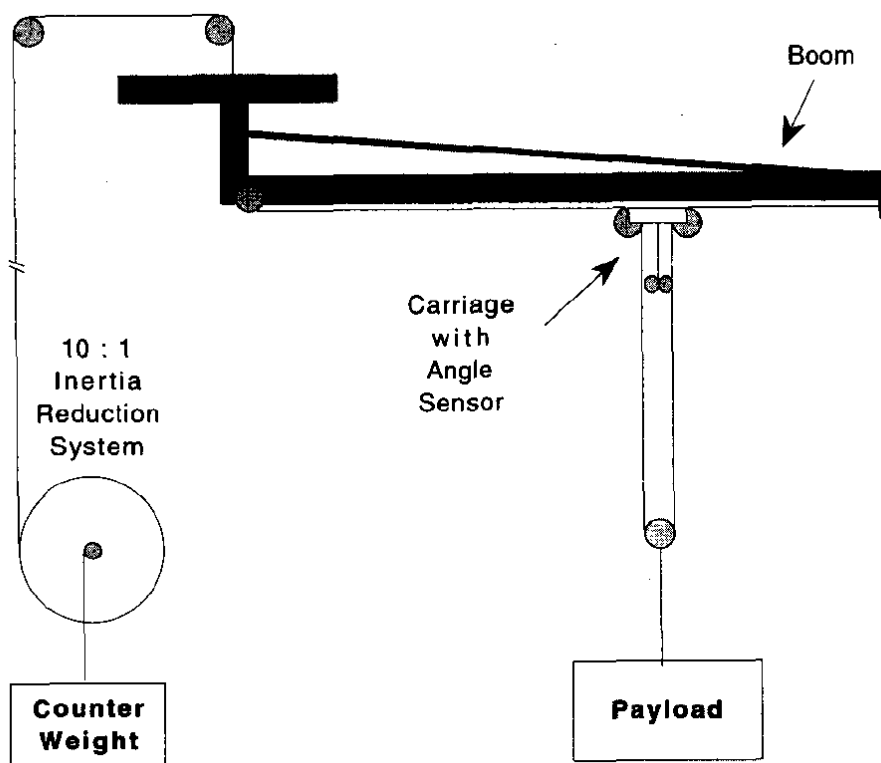
2.5.2. Mechanické metody

Existuje několik používaných způsobů pro vytváření kompenzační síly mechanickými metodami: například pomocí protizávaží, pomocí elektromotoru, nebo s použitím pneumatického systému. Výhodami takovýchto systémů kompenzace tíhové síly jsou především jejich relativní cenová nenáročnost (zvláště provozu) a dále také škálovatelnost – oproti ostatním zmíněným metodám nemají mechanické systémy kompenzace tíhové síly striktní objemové a hmotnostní omezení, lze je proto teoreticky přizpůsobit pro testování zařízení o téměř libovolném objemu a hmotnosti. Nevýhodou jsou parazitní vlivy tření a setrvačnosti kompenzačního systému, bodové působíště kompenzační síly a dále také nižší přesnost simulace mikrogravitace.[15]

Nejjednodušší přístup využívá protizávaží, jak je vidět například v práci „Mobility and manipulation of a light-weight space robot“[16] z roku 1992, kde je kompenzační systém tohoto typu využit pro testování vyvíjeného robotu v prostředí simulované mikrogravitace. Tato metoda spočívá v připojení testovaného objektu k protizávaží pomocí lana a soustavy kladek, které svojí hmotností na testovaný objekt vyvíjí kompenzační sílu. Jedná se o příklad pasivního kompenzačního systému. Další příklad pasivního kompenzačního systému je možné vidět na obrázku č. 2.1. Hlavní nevýhodou tohoto přístupu jsou parazitní setrvačné účinky silně ovlivňující dynamické vlastnosti systému, které protizávaží vnášejí do soustavy. Při přímém připojení protizávaží (o stejné hmotnosti jako testované těleso) k testovanému tělesu tak při simulaci mikrogravitace zdánlivě dvojnásobně vzroste hmotnost testovaného tělesa. Tomuto problému je možné částečně předejít připojením protizávaží k soustavě pomocí systému redukujícího setrvačné účinky protizávaží. Kromě protizávaží k systému přidávají své setrvačné účinky i kladky a lano, tyto účinky jsou ale

2. ZPŮSOBY SIMULACE MIKROGRAVITACE

zpravidla zanedbatelné vzhledem k hmotnosti testovaného tělesa. Další nevýhodou pasivních systémů kompenzace tíhové síly je zanedbatelné tření, které vzniká mezi kladkami a lanem a přispívá tak k dalšímu zneřádnění dynamiky. Z těchto důvodů tak, v případě pasivních systémů, nelze dosáhnout ideálních dynamických vlastností.[16][17]



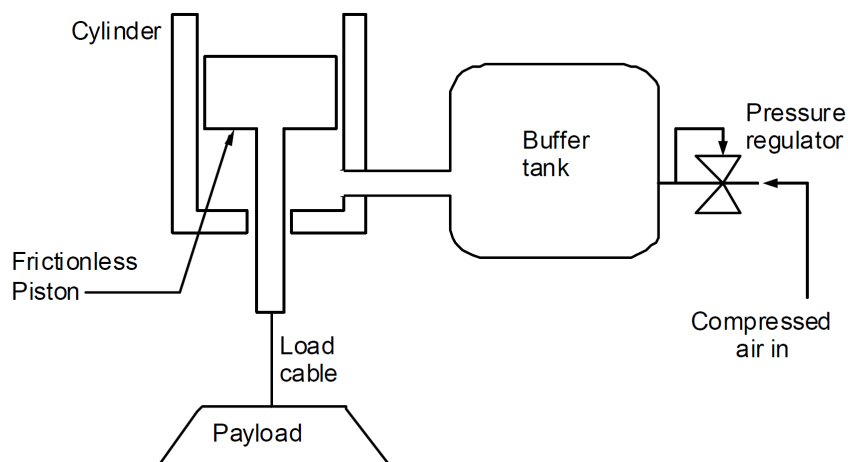
Obrázek 2.1: Příklad pasivního kompenzačního systému [17]

Dalším způsobem pasivní kompenzace tíhové síly je využití pneumatického systému, který je například použit v „Gravity-Offloading System for Large-Displacement Ground Testing of Spacecraft Mechanisms“[18]. V tomto případě je kompenzační síla vyvíjena pneumatickým pístem (obrázek č. 2.2) a řeší se tak problém s parazitními setrvačnými účinky metody využívající protizávaží. Další nežádoucí účinky v podobě tření v mechanických komponentech jsou však stále přítomny a zneřádnují tak kompenzaci tíhové síly.[18][19]

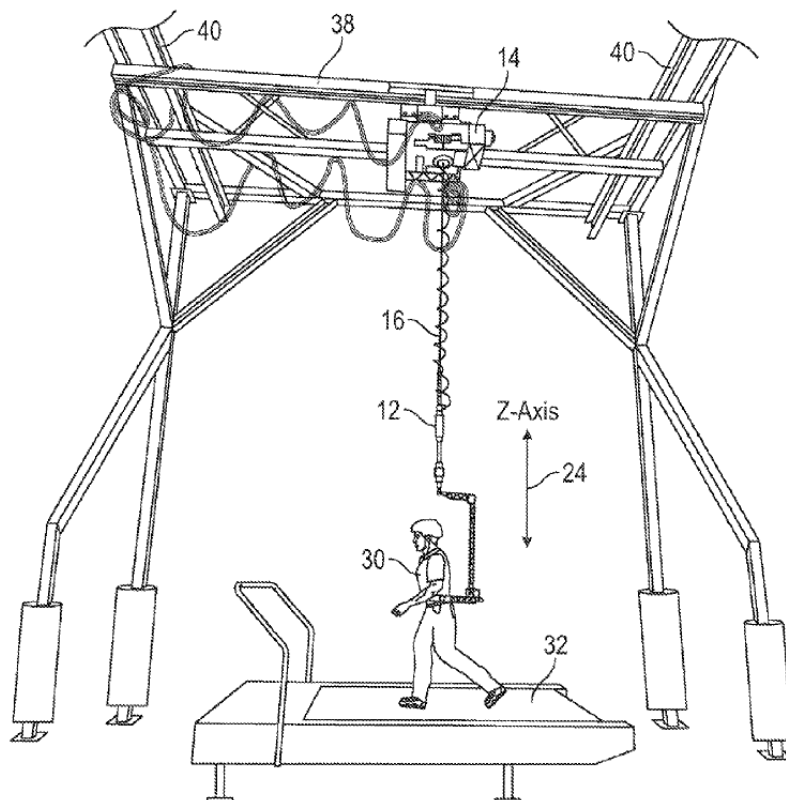
Aktivní řešení kompenzace tíhové síly představuje vyvíjení požadované kompenzační síly pomocí akčního členu – elektromotoru. Toto řešení je možné nalézt například v práci „Micro-G Emulation System Using Constant-Tension Suspension for a Space Manipulator“ [20] nebo v testovacím zařízení „ARGOS“[21] (Active Response Gravity Offload System) kosmické agentury NASA, které je možné vidět na obrázku č. 2.3. Aktivní kompenzace tíhové síly spočívá ve zpětnovazební smyčce, kde řídicí jednotka na základě zpětné vazby (kterou zajišťuje zpravidla senzor snímající sílu nadlehčující testované těleso) určuje pomocí řídicího algoritmu požadovanou hodnotu kompenzační síly a aktivně tak reguluje výstupní moment elektromotoru.[20][21][22]

Kombinací aktivního přístupu se zmíněnými pasivními metodami je možné kompenzovat nežádoucí vlivy setrvačnosti a tření pasivních metod jako například v práci „An Active

2.5. SYSTÉMY KOMPENZACE TÍHOVÉ SÍLY



Obrázek 2.2: Příklad kompenzačního systému využívajícího pneumatický píst [18]



Obrázek 2.3: Aktivního systém kompenzace tíhové síly ARGOS [22]

Z Gravity Compensation System“[17] z roku 1993. V této práci je pasivní systém kompenzace tíhové síly s protizávažím doplněn o elektromotor, vzniká tak hybridní systém se schopností kompenzace vlivů tření a setrvačnosti protizávaží pomocí aktivního řízení založeného na PI regulátoru a fuzzy logice.[17]

2. ZPŮSOBY SIMULACE MIKROGRAVITACE

Zde popsané principy mechanické kompenzace tíhové síly uvažují pohyb tělesa ve svislé ose, některé aplikace však vyžadují možnost volného pohybu i ve zbývajících 2 osách. Toho se dosahuje vytvořením posuvné vazby s minimálním třením v těchto osách, například pomocí lineárního ložiska. Dalším způsobem horizontálního pohybu může být aktivní posun v těchto osách pomocí elektromotorů, jako například ve zmíněném testovacím zařízení „ARGOS“.[15][21]

3. Projekt Zero-G

Projekt s pracovním označením Zero-G společnosti Sobriety s.r.o. je zaměřený na vývoj zmenšené verze testovacího zařízení, které má za účel simulaci dopadů ve stavu mikrogravitace a snížené gravitace. Zařízení na bázi kompenzace tíhové síly již bylo vytvořeno mnoho, i když mají všechna ve své podstatě stejný účel – simulaci mikrogravitace, jsou každé určené pro odlišnou aplikaci. Simulátor mikrogravitace, kterým se zabývá tato diplomová práce je svým návrhem a využitým principem uzpůsoben pro testování dopadů.

Hlavní předností principu simulátoru dopadu, kterým se zabývá tato diplomová práce je potenciálně vysoká hmotnostní kapacita. Jelikož většina existujících testovacích zařízení simulujících mikrogravitaci je silně omezená hmotností a objemem testovaného tělesa, není v některých případech možné provádět testy na větších součástech nebo soustavách v plném měřítku. V těchto případech je nutné testovat součást ve zmenšeném měřítku, nebo pouze analyzovat výpočetně, tyto metody však nemají plnou vypovídající hodnotu. Řešením tohoto problému je zařízení s dostatečnou kapacitou pro testování těles s vysokou hmotností.

Zde zkoumané testovací zařízení je velice komplexní systém, nejobecněji je však možné jej popsat jako spojení mechanického subsystému a elektronického subsystému, kterým se zabývá tato diplomová práce. Mechanickým aspektem tohoto testovacího zařízení se detailně zabývá práce „Mechanický návrh simulátoru dopadu za snížené gravitace“ [25].

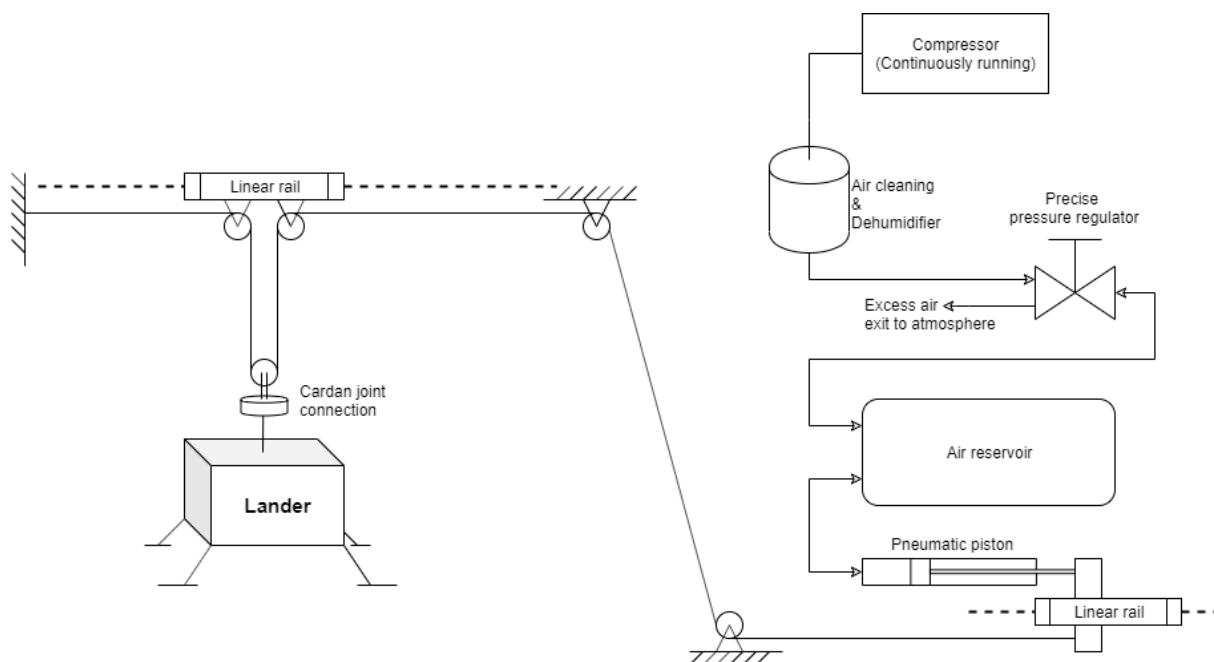
3.1. Princip

Navrhovaný simulátor pro testování dopadu ve snížené gravitaci je založen na principu kompenzace tíhové síly – na těleso je vyvíjena v každém okamžiku síla o velikosti tíhové síly ale opačného směru. Výsledkem je vzájemné vyrušení těchto dvou sil a těleso se tak zdánlivě ocitá ve stavu mikrogravitace. Pokud tato kompenzační síla bude menší než tíhová síla, je možné dosáhnout podmínek prostředí s libovolnou tíhovou silou (ale pouze menší než tíhová síla na Zemi). Avšak vytvořit tuto kompenzační sílu o neměnné hodnotě za všech podmínek, není triviální úkol.

V tomto řešení je kompenzační síla vyvíjena pomocí pneumatického pístu. Pneumatický píst je připojen ke vzduchové nádrži neboli vzdušníku, který má dostatečný objem na to, aby změny tlaku v soustavě způsobené zdvihem pneumatického pístu byly zanedbatelné. Díky tomuto klíčovému předpokladu – konstantnímu tlaku v pneumatické soustavě a tím pádem i konstantnímu tlaku v pneumatickém pístu – působí pneumatický píst v ideálním stavu konstantní silou bez ohledu na jeho polohu, nebo rychlost posuvu.

Jelikož je použitý pneumatický píst jednočinný a funguje pouze v tlačném režimu, je síla vyvíjená tímto pístem převedena na tah pomocí lineárního vedení, ke kterému je

přípevněné lano. Lano je dále přes soustavu kladek přivedeno k samotnému testovanému tělesu. Soustava je schematicky zobrazena na obrázku č. 3.1.



Obrázek 3.1: Schéma systému [24]

Pomocí této soustavy je teoreticky možné vyvinout na těleso konstantní sílu a docílit tak podmínek snížené gravitace nebo mikrogravitace. Síla vyvíjená pístem je úměrná tlaku v soustavě, pomocí regulace tlaku je tedy možné stanovit sílu vyvíjenou na těleso a určit tak velikost simulované tíhové síly.

Soustavu je dále možné modifikovat a přidat horizontální posuv pomocí kladek přípevněných na lineárním vedení, jak je možné vidět na obrázku č. 3.1, výhodou takovéto posuvné vazby je poté možnost provádění testování ve dvou osách. Příkladem testu vyžadujícího dvě osy je například simulace dopadu pod úhlem. Důsledkem této úpravy je avšak vytvoření převodového poměru 1:2 mezi silou, kterou působí pneumatický válec a silou působící na nadlehčované těleso. Z toho důvodu je také omezen posuv tělesa – při dané délce posunu pneumatického válce bude po této modifikaci svislý posun na straně tělesa pouze poloviční. Další nevýhodou je dodatečné tření způsobené dalšími kladkami a lineárním posuvem.

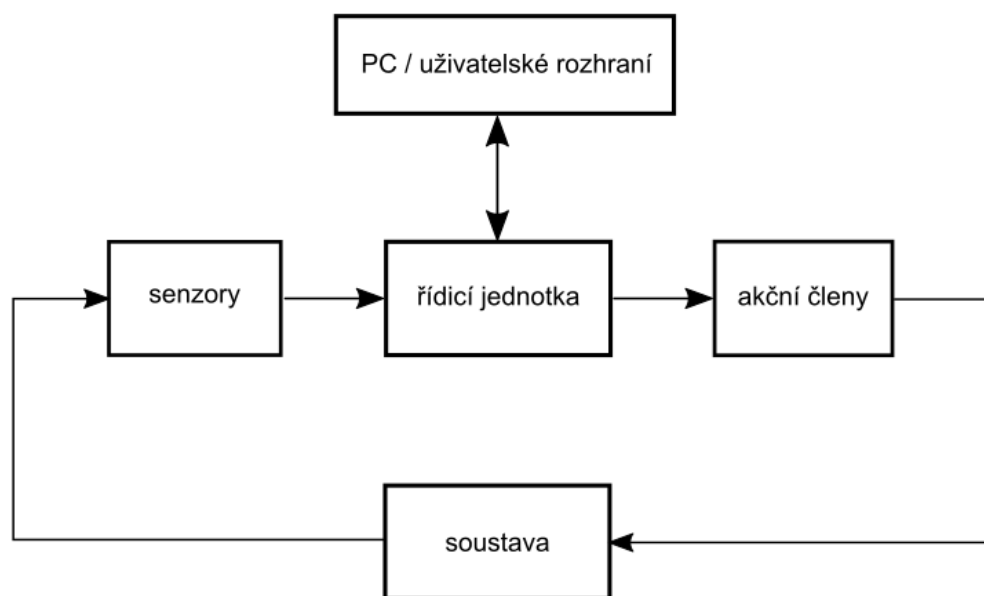
Pro účely testování dopadu ve snížené gravitaci je nutné na začátku experimentu udělit testovanému tělesu rychlost, kterou těleso narazí na povrch. Pro udělení této rychlosti bude systém vybaven elektromotorem, který bude působit silou proti pneumatickému pístu – bude těleso urychlovat směrem dolů. Jakmile bude dosaženo požadované rychlosti, motor je mechanicky odpojen a zkoumané těleso se pohybuje setrvačností k dopadové ploše. Mechanické odpojení je provedeno pomocí lana, kterým motor urychluje těleso. Při dosažení cílové rychlosti motor zpomalí, lano se v důsledku toho povoluje a motor přestává působit na soustavu. Urychlení v horizontální ose může být dosaženo analogicky.

3.2. POŽADAVKY NA ELEKTRONICKÝ SUBSYSTÉM

Prvky mechanické soustavy jako kladky, lineární vedení, lano nebo pneumatický válec vnášejí do systému vlivy nežádoucího tření a tlumení, toto je jedna z nevýhod systémů mechanické kompenzace tíhové síly. Z toho důvodu nebude simulace snížené gravitace nikdy ideální. Jednou z možností řešení tohoto problému je aktivní kompenzace třecích sil pomocí elektromotoru, který na základě zpětné vazby ze senzorů minimalizuje odchylku kompenzační síly od stanovené hodnoty. Aktivní kompenzací se ovšem tato diplomová práce nebude zabývat.

3.2. Požadavky na elektronický subsystém

Navrhovaný simulátor dopadu je komplexní mechatronický systém a vyžaduje pro svoji funkci řadu senzorů a akčních členů, které jsou součástí elektronického subsystému. Elektronický subsystém má na starosti snímání dat během dopadového testu, monitorování měřených veličin dopadového simulátoru a dále také řízení samotného experimentu, do čehož spadá například zajištění tělesa v počáteční pozici před začátkem experimentu nebo urychlení tělesa na požadovanou počáteční rychlost na začátku experimentu. Obecné schéma navrhovaného subsystému je na obrázku č. 3.2.



Obrázek 3.2: Obecná struktura navrhovaného systému [26]

Pro měření tlaku v pneumatickém systému jsou vyžadovány senzory tlaku, pomocí kterých bude možné před začátkem experimentu nastavit požadovanou hodnotu tlaku a tím pádem i tlaku přímo úměrnou kompenzační sílu, která bude působit na testované těleso. Senzory tlaku budou také využity pro monitorování a kontrolování hodnot tlaku v několika místech systému během testování, tyto hodnoty budou pouze zobrazovány a zaznamenávány a nebudou sloužit jako zpětná vazba v reálném čase pro jinou součást

systému. Z tohoto důvodu není požadována vysoká vzorkovací frekvence, pro potřeby monitorování postačí nižší vzorkovací frekvence, řádově v jednotkách Hz. Na rozdíl od hodnoty vzorkovací frekvence, požadavek na přesnost bude náročnější, aby bylo možné detekovat například jemné změny tlaku z důvodu netěsností. Díky vyšší přesnosti senzorů tlaku bude také možné nastavit s větší přesností počáteční tlak a zaručit tak dobrou opakovatelnost experimentů.

Monitorování síly v laně neboli kompenzační síly, která nadlehčuje těleso, bude zajištěno tenzometrem umístěným mezi testovaným tělesem a lanem. V ideálním případě bude hodnota síly měřená tenzometrem konstantní a bude mít hodnotu rovnou požadované odlehčovací síle. Zpětná vazba z tenzometru bude tedy vypovídat o kvalitě kompenzace gravitační síly. Pro umožnění pozorování dynamických dějů při změnách odlehčovací síly je požadovaná hodnota vzorkovací frekvence měření síly 1000 Hz a více.

Pro účely analýzy dopadu ve snížené gravitaci bude sloužit akcelerometr připevněný na samotném zkoumaném tělese. Data z akcelerometru nebudou sloužit jako zpětná vazba v reálném čase pro jiné části systému, budou pouze zobrazovány a zaznamenávány a budou použity pro zpětnou analýzu dopadu.

Další požadavek na elektronický subsystém je snímání polohy pneumatického pístu, která odpovídá poloze zkoumaného tělesa ve svislé ose (při zanedbání protažení lana a za předpokladu napnutí lana). Toto bude uskutečněno pomocí lineárního enkodéru na lineárním vedení, které je připevněno k pístnici pneumatického pístu. Stejně odměřování vzdálenosti bude možné aplikovat i na horizontální posun, kde lineární enkodér bude připevněn k posuvné kladce na lineárním vedení. Z měřené polohy bude dále řídicí jednotkou odhadována rychlost tělesa. Záznam poloh bude sloužit zejména pro potřeby zpětné analýzy průběhu experimentu.

Hlavními akčními členy elektronického subsystému bude dvojice elektromotorů, které budou mít za úkol urychlení testovaného tělesa v horizontální a vertikální ose při začátku experimentu. Dalším akčním členem bude solenoidová západka, která bude držet těleso v počáteční poloze před začátkem experimentu. V případě simulování mikrogravitace není západka zapotřebí, jelikož těleso bez vnějších vlivů setrvá ve své poloze, v případě simulace snížené gravitace je avšak nutné před přípravou experimentu testované těleso zabezpečit před posunem.

Jednou z nejdůležitějších součástí elektronického subsystému bude řídicí jednotka, která bude mít za úkol příjem dat ze senzorů, zpracování dat a vysílání dat pro řízení akčních členů. Řídicí jednotka musí disponovat dostatečným výpočetním výkonem pro dosažení požadovaných frekvencí zpracování dat, které byly přibližně vytyčeny v této kapitole. Dalším požadavkem na řídicí jednotku je možnost připojení senzorů s různými rozhraními. Jelikož se jedná o projekt ve fázi vývoje, je pravděpodobné, že nastanou nepředpovídané změny a nelze proto zcela předpovídat všechna potřebná rozhraní. Podobně je dále nutné počítat se softwarovými změnami, například komunikačních protokolů. Z tohoto důvodu je jedním z požadavků na řídicí jednotku její flexibilita a univerzálnost jak v ohledu hardwaru, tak i softwaru. Dalším důležitým požadavkem ohledně řídicí jednotky je schopnost řízení a zpracovávání dat v reálném čase, na které závisí správné načasování zásahů akč-

3.2. POŽADAVKY NA ELEKTRONICKÝ SUBSYSTÉM

ních členů a správná funkce celého testovacího zařízení. Příkladem je proces urychlování tělesa, kde je správné načasování zásahů do systému kritické pro dosažení cílové rychlosti v okamžiku dopadu. Cena řídicí jednotky je také důležitou proměnnou, přirozeně je preferováno co nejlevnější vhodné řešení.

Nezbytnou součástí elektronického subsystému je také uživatelské rozhraní, které bude v tomto případě realizováno s pomocí PC. Uživatelské rozhraní bude umožňovat uživateli kontrolovat stav systému, monitorovat měřené hodnoty v reálném čase a také ovládat průběh experimentu. Požadovanými vlastnostmi grafického uživatelského rozhraní na PC je poskytnout uživateli intuitivní a rychlý způsob interakce s testovacím systémem. Grafické uživatelské rozhraní bude nabízet možnost zobrazení dat z libovolných senzorů systému v reálném čase, jejich analýzu a také možnost jejich uložení. Konkrétní způsob implementace uživatelského rozhraní pak dále závisí z velké části na volbě řídicí jednotky.

4. Možnosti řízení a sběru dat v podobných aplikacích

Projekt Zero-G je možné charakterizovat jako systém skládající se z mechanické soustavy, ze senzorů snímajících vybrané veličiny, z akčních členů působících na mechanickou soustavu a dále také z řídicí jednotky pro vyhodnocování měřených dat a vytváření řídicího signálu pro akční členy. Jelikož cílem této diplomové práce je návrh subsystému pro řízení a sběr dat této soustavy, bude se tato kapitola zabývat běžně používanými řešeními.

Konkrétně budou v této kapitole popsány tři možnosti řízení, které by bylo možné uplatnit pro systém, o kterém pojednává tato diplomová práce, nebo pro řízení strukturně podobných mechatronických systémů. Jako tyto tři způsoby řízení byly vybrány PLC, vestavěné systémy (neboli řízení vytvořené na základě mikrokontroleru) a nakonec řízení s pomocí osobního počítače. U každého z těchto přístupů bude popsána charakteristika a hlavní vlastnosti.

4.1. PLC

Jedním dnes z nejpoužívanějších řešení řízení elektromechanických systémů v průmyslu je PLC (Programmable Logic Controller) neboli programovatelný logický automat. PLC je řídicí jednotka navržena pro řízení průmyslových procesů a jejich automatizaci, je možné jej také chápat jako řídicí jednotku, která má software a hardware uzpůsobený pro řízení průmyslových aplikací.[27][28][29]

Vývoj PLC začal v 70. letech minulého století. Hlavní motivací pro vznik PLC byl automobilový průmysl, ve kterém byly do té doby pro řízení výrobních procesů používány systémy na bázi relé. Podstatnou nevýhodou těchto systémů byla vysoká časová náročnost v případě nutnosti provedení změn v řízení, jelikož logika řízení byla určena fyzickým propojením jednotlivých komponent. Požadavkem byl tedy nový typ řídicího systému na polovodičové bázi, umožňující provádění rychlých změn řízení v případech modifikace výrobních procesů bez nutnosti hardwarových změn. Pro ulehčení přechodu z reléových systémů na PLC systémy byl pro PLC navržen nový typ programovacího jazyka, který byl inspirován logickými diagramy zapojování relé. [30][29]

Hlavní součástí PLC, jako u jiných příkladů výpočetní techniky, je mikroprocesor. Slouží pro vykonávání programu definovaného uživatelem, který je uložený v paměti PLC. Pro ukládání programu se používá paměť flash, EEPROM (neboli elektronicky vymazatelná a programovatelná paměť pro čtení), nebo jiný typ paměti, který program zachová i v případě výpadku napájení. Program je v paměti uložen v binární formě, přeložený z vysokoúrovňového jazyka, pomocí kterého byla řídicí logika uživatelem definována.[29]

4.1. PLC

Systémy PLC je možné programovat pomocí řady programovacích jazyků. Jsou to například: SFC – sequential function chart (vývojové schéma), FBD – function block diagram (schéma funkčních bloků), LD – ladder-logic diagram (reléové schéma), ST – structured text, IL – instruction list (posloupnost instrukcí), nebo více konvenční jazyky jako C, C++ a další. Každý ze zmíněných způsobů programování má svoje výhody a nevýhody, avšak nejrozšířenější je programování PLC pomocí LD – reléových schémat (někdy také nazývaných liniová schémata). Tento způsob programování napodobuje způsob značení zastaralých reléových logických obvodů a je to nejstarší způsob programování PLC. Všechny nástroje pro programování PLC se dají rozdělit do dvou kategorií – grafické a textové nástroje. Do grafických způsobů programování PLC spadají LD, FBD, SFC a do textových například IL, ST, C, C++. Obecně, grafické nástroje jsou více přehledné a intuitivní a umožňují rychlé pochopení řídicí logiky, textové nástroje nabízí více konvenční způsob zápisu programu a jsou výhodnější pro určité typy aplikací.[29]

Velmi důležitým prvkem systému PLC jsou jeho vstupní a výstupní rozhraní, jelikož hlavní funkcí PLC je snímat měřené veličiny a vytvářet výstupy pro řízení akčních členů. Jako vstup do PLC může sloužit řada zařízení, například pro interakci uživatele s řídicím systémem je možné použít spínače, klávesnice, atd... Pro snímání fyzikálních veličin a stavů řízeného systému jako vstup do PLC slouží senzory. Akční členy, které jsou často řízené pomocí PLC jsou třeba elektromotory, solenoidové akční členy, relé, atd... Existuje mnoho typů senzorů, akčních členů a periférií, důležitý je však způsob, jakým dané zařízení komunikuje. Typy výstupů a vstupů je možné rozdělit do několika kategorií: logická (binární) hodnota, zde rozlišujeme pouze dva stavy. Tento typ byl používán od počátku PLC, jelikož na stejné binární bázi fungovaly předcházející reléové systémy. Dalším typem „komunikace“ se senzory a akčními členy je analogová hodnota neboli hodnota, která může nabývat velikosti v určitém rozsahu. V tomto případě musí PLC disponovat ADC převodníkem pro převedení analogového signálu ze snímače na digitální a DAC převodníkem na generování analogového signálu v případě řízení akčního členu analogovým vstupem. Některé senzory a akční členy mohou používat pro komunikaci průmyslové sběrnice a protokoly, jako například CAN bus, Modbus, Profibus nebo DeviceNet. PLC musí být v tomto případě vybaveno hardwarem a softwarem podporující dané standardy. Kromě komunikace se senzory mohou být komunikační rozhraní také používány pro programování PLC.[29][30]

PLC je možné rozdělit do několika hlavních typů. Kompaktní PLC spojuje hlavní funkce PLC systému do jednoho zařízení. Modulární PLC je opakem kompaktního, jednotlivé prvky PLC (jako například výpočetní část, vstupní rozhraní, výstupní rozhraní a jiné periferie) jsou umístěny ve vlastních modulech, které jsou propojeny a komunikují pomocí společné sběrnice. Výhoda modulárního systému je vyšší flexibilita a potenciálně velké množství vstupů a výstupů oproti kompaktní variantě. Dalším typem PLC je PLC ve formě rozšiřující karty do PC. Toto zařízení je samostatná jednotka komunikující s PC pomocí PCI sběrnice, její chod je však nezávislý na PC.[28]

PLC je běžným příkladem řídicího systému, který disponuje řízením v reálném čase. Tato vlastnost je velmi důležitá pro řízení mnoha průmyslových i jiných systémů, pro které je rychlost a předvídatelnost odezvy řízení kritická. PLC uskutečňuje řízení v reálném čase

pomocí operačních systémů reálného času, které prioritně dbají na dodržování časových limitů při vykonávání uživatelem definovaného programu.[31]

Závěrem, PLC můžeme chápat jako systém uzpůsobený pro průmyslové prostředí určený pro zpracování hodnot ze snímačů různých druhů a vytváření řídicího výstupu pro akční členy pomocí uživatelem definovaného programu. Hlavní důvody pro používání PLC systémů v průmyslu jsou jejich robustnost, spolehlivost a možnost rychlé implementace, naopak jejich nevýhodou je vyšší cena za jednotku oproti jiným řešením.[32][33]

4.2. Vestavěné systémy

Vestavěné systémy (anglicky „embedded systems“) jsou řídicí systémy založené na mikroprocesoru nebo mikrokontroleru, které jsou navrženy pro konkrétní aplikaci a mají specifický účel. Tyto systémy mohou být fyzicky vestavěné v dané aplikaci, odtud tedy pochází název „vestavěné systémy“ a na rozdíl od jiných typů řízení, které jsou umístěny v samostatném zařízení, jsou vestavěné systémy často integrální součástí řízeného systému. Vestavěné systémy se v různých podobách vyskytují ve většině běžných přístrojů, jako například v mikrovlnných troubách, síťových routerech, automobilech a mnoha dalších zařízeních. [27][34]

Vestavěné systémy jsou na rozdíl od jiných výpočetních a řídicích systémů navrhovány tak, aby byly právě splněny výkonnostní požadavky dané aplikace a aby disponovaly jenom těmi hardwarovými periferiemi, které jsou pro řízený systém potřebné. Podobným způsobem je navrhován i software pro vestavěné systémy – jelikož mají velmi specifický účel je možné vytvořit nenáročný specializovaný software, který je možné vykonávat i na systémech s omezenými hardwarovými prostředky. Vestavěné systémy jsou navrhovány daným aplikacím „na míru“ podle požadavků jako jsou cena, rozměry, spotřeba energie a další.[26][34]

Jako jeden z prvních vestavěných systému bývá označován naváděcí počítač Autonetics D-17 z roku 1966, který sloužil pro navádění střely Minuteman II, jelikož jako první využíval procesor a paměť ve formě integrovaných obvodů. V 70. letech 20. století začaly být mikroprocesory dostupné a začaly se používat pro ovládání průmyslových procesů, vznikaly tak první průmyslové vestavěné systémy a od té doby se vestavěné systémy staly jednou z nejrozšířenějších řídicích technologií.[35]

Vestavěný systém může být vytvořen nejčastěji s použitím mikroprocesoru, nebo mikrokontroleru. Mikrokontroler na rozdíl od mikroprocesoru integruje navíc v jednom integrovaném obvodu další komponenty jako jsou například paměti RAM a ROM nebo také vstupní a výstupní periferie. Integrované periferie umožňují mikrokontroleru zjednodušit vestavěný systém díky odstranění externích periférií. V případě vytváření vestavěného systému s použitím mikrokontroleru je pak bez dalších komponent možné využití jeho vestavěných funkcí čímž mohou být například časovače, AD, DA převodníky, nebo DMA a běžné jsou také komunikační rozhraní jako RS232, I2C, SPI nebo USB. Pro vytváření

4.2. VESTAVĚNÉ SYSTÉMY

vestavěných systémů je kromě mikroprocesorů a mikrokontrolerů další možností například FPGA (field programmable gate array) neboli programovatelné hradlové pole a nebo také CPLD (complex programmable logic device) neboli komplexní programovatelný logický obvod. Na rozdíl od mikroprocesorů a mikrokontrolerů, kde uživatel definuje software, FPGA a CPLD dovoluje uživateli definovat i hardwarovou úroveň logického obvodu, což zvyšuje flexibilitu těchto integrovaných obvodů. Tato vlastnost umožňuje konfiguraci těchto prvků přímo na míru dané aplikaci. FPGA a CPLD logické obvody také často nabízejí možnost modifikovat hardware za běhu, což umožňuje těmto zařízením realizaci různých rozhraní nebo algoritmů v závislosti na aktuální potřebě aplikace. Poslední možností pro vestavěné systémy, která zde bude zmíněna, je použití DSP (digital signal processor) neboli digitálního signálového procesoru. Tento integrovaný obvod se podstatou podobá mikrokontroleru, oproti tomu má však hardware uzpůsobený pro zpracování signálu, je proto vhodný pro použití ve vestavěných systémech například pro zpracování zvuku.[36]

Pro propojení se snímači, akčními členy a dalšími komponenty musí být vestavěný systém vybaven patřičnými hardwarovými perifériemi. V případě vestavěných systémů založených například na mikrokontroleru je možné využít integrovaných periférií mikrokontroleru. V případě, kdy potřebné periferie nejsou integrované, je nutný dodatečný externí hardware. Mezi periferie, které je díky mikrokontroleru možné přímo použít, se běžně řadí sériová komunikační rozhraní, digitální vstupy a výstupy, nebo ADC a DAC převodníky. Ne ve všech případech jsou logické napěťové úrovně stejné mezi mikrokontrolerem a zařízeními, se kterými interaguje, v těchto případech je nutné hardwarové přizpůsobení napěťových úrovní. Jelikož procesory a mikrokontrolery jsou schopné pracovat pouze s logickými signály, pro řízení akčních členů a zařízení, které vyžadují vyšší proudový odběr, je nezbytné použití například tranzistorů, pomocí kterých logické obvody mohou spínat tyto silové obvody. Tímto způsobem lze řídit například relé nebo solenoidové akční členy, při zapojení tranzistorů do uspořádání H-můstku a PWM modulaci řídicího signálu, je možné ovládat vestavěným systémem například stejnosměrný motor. V závislosti na konkrétních požadavcích jsou do vestavěného systému implementovány periferie, které aplikace vyžaduje, vzhledem ke způsobu návrhu vestavěných systémů “na míru“ je možné vytvořit vestavěný systém kompatibilní s libovolnými rozhraními, typy komunikace apod.[37][36][38]

Vestavěné systémy jsou zpravidla programovány v nízkourovňových programovacích jazycích, jsou to nejčastěji například jazyky C a assembler. Použití nízkourovňových jazyků nabízí precizní kontrolu nad programovaným logickým obvodem, assembler jako nejnižší programovací jazyk může být z toho důvodu použit pro programování kritických sekcí programu. Jazyk C nabízí na rozdíl od assembleru vrstvu abstrakce a je proto více uživatelsky přívětivý. Vysokourovňové jazyky nejsou vhodné pro programování vestavěných systémů a nejsou proto pro tento účel obvykle používány. Jeden z důvodů pro jejich nevhodnost je například vysoká náročnost na paměť.[39][40]

Častým požadavkem řízených aplikací na řídicí vestavěný systém je řízení v reálném čase. Řízení v reálném čase pomocí vestavěných systémů lze uskutečnit využitím existujících RTOS (real-time operating system) – operačních systémů reálného času. Další možností je vytvoření vlastního softwaru pro tento účel. V případě využití operačního systému reálného času je řídicí proces rozdělen do podprocesů a vláken, které operační systém řídí tak,

aby byly dodrženy stanovené časové limity. Použití RTOS je zpravidla rychlejší a spolehlivější způsob vytváření řídicího softwaru reálného času, oproti tomu vytváření vlastního real-time softwaru s sebou nese více úskalí a jeho vývoj může být časově náročnější.[41][36]

4.3. Měření a řízení pomocí PC

Další z možností měření a řízení je využití běžného osobního počítače. V takovém případě osobní počítač slouží ke zpracování dat ze zkoumaného systému, jejich ukládání, vytváření řídicího signálu a také jako uživatelské rozhraní. Dodatečný hardware je zpravidla nezbytný jako rozhraní mezi PC a vstupy/výstupy řízené soustavy, s takovýmto hardwarem pak osobní počítač komunikuje pomocí některé ze svých sběrnic. Využití osobních počítačů je časté u většiny řídicích systémů, například ve funkci uživatelského grafického rozhraní pro řízený proces, pro konfiguraci dané řídicí jednotky, nebo lze PC použít na vyšší úrovni řídicího systému pro dohled nad řídicí jednotkou, v těchto aplikacích však osobní počítač nevykonává samotné řízení. Tato sekce se zabývá řídicími systémy, kde osobní počítač přímo zastává roli řídicí jednotky.[42]

Řešení měření a řízení pomocí PC nabízí řadu výhod, jedna z předních výhod je flexibilita a přizpůsobitelnost tohoto řešení, umožňující uživateli jednoduše a rychle modifikovat řídicí systém podle měnících se požadavků aplikace. Popularita a všudypřítomnost osobních počítačů činí pak tuto metodu měření a řízení více přístupnou a uživatelsky přívětivou ve srovnání s ostatními metodami, k uživatelské přívětivosti přispívá také uživatelské rozhraní, kterým disponuje každý osobní počítač. Oproti řízení s vestavěnými systémy, které jsou zpravidla navrhovány s minimem hardwarových prostředků, disponují osobní počítače značně vyšším výpočetním výkonem, tato vlastnost tak umožňuje řídicím systémům na bázi PC provádět složitější procesy zpracování naměřených dat, nebo například použití více náročných řídicích algoritmů. Jedním z příkladů je použití osobního počítače v aplikacích vyžadující zpracování obrazu, zde je použití PC vhodné z důvodu množství dostupné paměti.[42]

Jak již bylo zmíněno, měření a řízení pomocí PC vyžaduje dodatečné vstupně-výstupní zařízení jako rozhraní mezi řízeným systémem a PC. V závislosti na použité sběrnici (pro komunikaci mezi PC a vstupně-výstupním zařízením) mohou mít tato zařízení formu rozšiřujících karet v případě sběrnic jako například PCI, PCI Express nebo ISA/EISA. Vstupně-výstupní rozšiřující karty existují také v podobě modulů pro platformu PXI, technologii založené na sběrnici PCI, kde jsou rozšiřující moduly umístěny v externím šasi. PXI systémy nabízí možnost vytvoření modulární instrumentace pro PC, která je vhodná pro náročné průmyslové měřicí nebo řídicí aplikace. Další formou vstupně-výstupních zařízení mohou být externí zařízení komunikující s PC pomocí rozhraní USB. Existuje mnoho typů vstupně-výstupních zařízení, běžným příkladem jsou multifunkční vstupně-výstupní karty, jsou to rozšiřující karty pro univerzální použití, které disponují základními funkcemi jako analogové vstupy a výstupy, digitální vstupy a výstupy, nebo časovače. Kromě multifunkčních karet jsou dostupné také moduly se specifickým zaměřením, jako například moduly určené pro měření napětí, proudu, tlaku, teploty, síly, moduly

4.3. MĚŘENÍ A ŘÍZENÍ POMOCÍ PC

pro zpracování zvuku nebo vibrací, moduly pro komunikaci s průmyslovými sběrnici a nebo moduly určené pro aplikace počítačového vidění. Velká škála dostupných vstupně-výstupních zařízení umožňuje využití osobního počítače jako řízení pro téměř jakoukoliv aplikaci.[43][44]

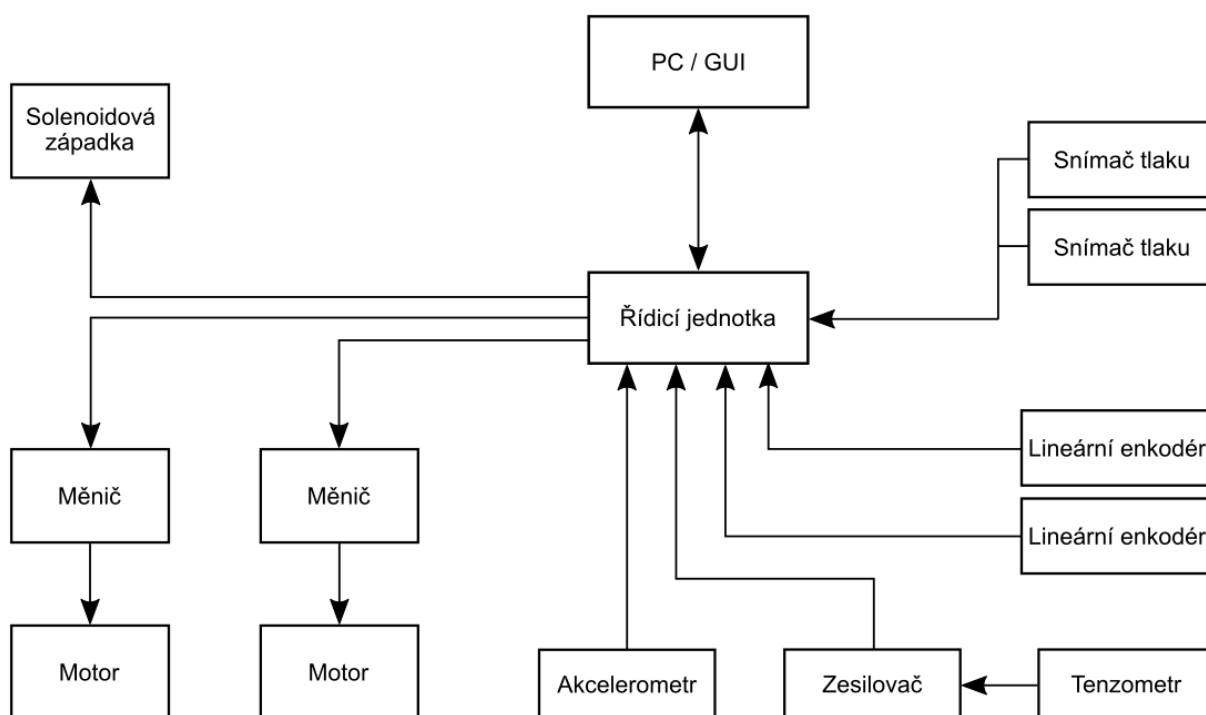
Pro programování řídicích systémů založených na osobním počítači mohou být použita grafická vývojová prostředí, jako například běžně používané prostředí LabView (ve spojení se vstupně-výstupními zařízeními od firmy National Instruments), toto prostředí nabízí kromě grafické definice programu také jednoduchý způsob vytváření grafického uživatelského rozhraní. Dalším grafickým prostředím použitelným pro programování PC řídicích systémů je například prostředí Simulink – grafické rozšíření prostředí Matlab. Z více tradičních textových programovacích jazyků jsou pro programování systémů tohoto typu používány například jazyky C nebo C++.[45]

Jednou z nevýhod řídicích systémů založených na osobním počítači je nevhodnost pro aplikace vyžadující řízení v reálném čase, tato vlastnost vychází z nedeterministické odezvy běžných operačních systémů. Tento problém je možné řešit použitím softwaru využívajícího jádra reálného času, kdy je samotný řídicí program vykonáván na nižší úrovni s větší předvídatelností a s vyšší prioritou. Další možnost je využití operačního systému reálného času na řídicím osobním počítači. Těmito způsoby lze překonat nedeterministické vlastnosti osobních počítačů a uplatnit řízení pomocí PC i na systémy vyžadující řízení v reálném čase.[46]

I když je použití běžného osobního počítače možné, v některých aplikacích (zejména průmyslových) je vyžadována vyšší robustnost a spolehlivost samotného hardwaru. Běžný počítač má pohyblivé části jako například pevné disky, které jsou ze své podstaty poruchovější oproti zařízením bez pohyblivých částí. Běžné počítače dále mohou být náchylné na rušení v případě blízkých zdrojů rušení, jakým může být například měnič. Z těchto důvodů jsou pro náročné aplikace dostupné průmyslové počítače, zaměřující se na odolnost a spolehlivost hardwaru ze kterého se PC skládá. Jejich výhodami jsou zmíněná odolnost, například proti prachu, vibracím, elektromagnetickému rušení, nebo změnám teploty, díky tomu mohou dosahovat vyšší spolehlivosti. Nevýhodou průmyslových počítačů oproti běžným PC je vyšší cena.[46][47]

5. Návrh elektronického subsystému

Návrh elektronického subsystému vychází z požadavků popsanych v kapitole č. 3 v sekci „požadavky na elektronický subsystém“, kde byla popsána vyžadovaná funkcionalita tohoto subsystému. Tyto požadavky lze graficky shrnout zjednodušeným schématem elektronického subsystému, které je možné vidět na obrázku č. 5.1.



Obrázek 5.1: Schéma elektronického subsystému

Tato kapitola bude pojednávat o jednotlivých prvcích systému, o využitých komponentech a jejich hardwarové a softwarové integraci do systému a také o elektronickém subsystému jako celku. V rámci této kapitoly bude dále zmíněno mimo jiné například napájení systému a hardware s tím spojený. Kromě samotného teoretického návrhu bude popsána také realizace tohoto systému včetně úskalí a zvolených řešení.

5.1. Řídicí jednotka

Byly uvažovány 3 možné přístupy při volbě řídicí jednotky, které byly charakterizovány v kapitole „Možnosti řízení a sběru dat v podobných aplikacích“, jsou to tedy řízení pomocí PLC, řízení pomocí vestavěného systému založeného na mikrokontroleru, nebo řízení založené na osobním počítači. Pro navrhovaný elektronický subsystém byl vybrán způsob

5.2. VÝVOJOVÉ PROSTŘEDÍ

řízení a sběru dat pomocí vestavěného systému vytvořeného na základě mikrokontroleru. Toto řešení bylo zvoleno z důvodu nízké cenové náročnosti samotného hardwaru, která je nižší ve srovnání s pořizovacími náklady PLC systémů nebo vstupně-výstupního hardwaru v případě řídicího systému založeného na osobním počítači. Dalším důvodem pro volbu řízení pomocí vestavěného systému je flexibilita, kterou vestavěné systémy nabízejí, jelikož vestavěný systém bude vytvořen „na míru“ požadavkům, které byly vytyčeny aplikací. V případě změn požadavků během vývoje je možné dále modifikovat a přizpůsobovat vestavěný systém a jeho periferie pro splnění nových požadavků. Nevýhodou tohoto přístupu je vyšší časová náročnost na vývoj z důvodu samotného vytváření vestavěného systému.

5.1.1. STM32F767ZI

Jako základ vestavěného systému byl zvolen mikrokontroler STM32F767ZI a pro vývoj pomocí tohoto mikrokontroleru byla použita vývojová deska Nucleo-F767ZI. Tato vývojová deska disponuje zmíněným mikrokontrolerem STM32F767ZI, nástrojem ST-LINK V2 pro programování a ladění tohoto mikrokontroleru. Dále vývojová deska nabízí například také krystal 32,768 kHz, rozhraní USB a Ethernet, dvě tlačítka a tři LED diody pro účely vývoje. Samotný mikrokontroler je založen na 32-bitovém jádře ARM Cortex-M7 s maximální frekvencí 216 MHz a disponuje mimo jiné 2 MB flash pamětí, 512 kB SRAM pamětí, řadou krystalů (4-26 MHz, 16 MHz, 32 kHz), třemi 12-bitovými DA převodníky, dvěma 12-bitovými AD převodníky, funkcí DMA pro až 16 kanálů, 168 vstupně-výstupními porty s možností vnějšího přerušení, 18 časovači, 28 komunikačními rozhraními (jako například I2C, UART/USART, SPI nebo CAN), jednotkou pro výpočet CRC, analogovým generátorem náhodných čísel a mnohým dalším.[48][49]

Při vytváření softwaru pro mikrokontroler byla využita knihovna HAL (Hardware Abstraction Layer) neboli vrstva abstrakce hardwaru pro mikrokontrolery řady STM32F7, poskytující API, které uživateli umožňují jednodušší práci s periferiemi mikrokontroleru. Příkladem jsou API pro komunikační periferie, kde je díky knihovně HAL umožněno inicializovat a konfigurovat komunikační sběrnice voláním několika příslušných funkcí. Kromě API pro komunikační periferie, HAL obsahuje další funkce, například pro práci s DMA, časovači aj.[50]

5.2. Vývojové prostředí

K vývoji softwaru pro mikrokontroler byla využita vývojová prostředí STM32CubeMX a Atollic TrueSTUDIO.

STM32CubeMX je grafický nástroj umožňující jednoduchou inicializaci a konfiguraci STM32 mikrokontrolerů a jejich periferií. Díky knihovně STM32 mikrokontrolerů a vývojových desek obsažených v tomto programu, jsou v STM32CubeMX zahrnuty informace o daném mikrokontroleru, jeho periferiích, vstupech, výstupech a mnohém dalším, které by

při „manuální“ inicializaci bylo nutné získat z manuálu příslušného čipu. Díky zahrnutým informacím o použitém mikrokontroleru a jeho perifériích STM32CubeMX nabízí v grafickém rozhraní relevantní možnosti pro daný mikrokontroler a umožňuje detekovat neplatné konfigurace. Další z funkcí tohoto programu je možnost nastavení frekvencí mikrokontroleru pomocí intuitivního frekvenčního schématu, do kterého uživatel zadá požadované frekvence a program automaticky systém přizpůsobí (např. změnou hodnot děliček) pro dosažení těchto frekvencí. Po dokončení nastavení v grafickém rozhraní STM32CubeMX vytvoří projekt ve vybraném vývojovém prostředí a vygeneruje inicializační zdrojový kód pro daný mikrokontroler. Tímto způsobem je možné jednoduše inicializovat komunikační rozhraní, časovače, přerušení, DMA, frekvence pro různé komponenty mikrokontroleru a další.[51]

Atollic TrueSTUDIO for STM32 je vývojové prostředí určené pro vývoj softwaru pro mikrokontrolery STM32. Je založeno na open-source vývojovém prostředí Eclipse a nabízí běžné nástroje pro vývoj softwaru pro mikrokontrolery. Je tedy pomocí něj možné editovat kód, kompilovat jej a nebo také ladit vyvíjený program.[52]

5.3. Obecná struktura C programu

Pro efektivní využití dostupného výkonu mikrokontroleru STM32F767ZI byla vytvořena obecná struktura C programu, po vzoru které byly softwarově implementovány jednotlivé komponenty. Tato struktura je založena na co možná nejvyšším využití nástrojů jako například přerušení nebo DMA pro asynchronní vykonávání více úkonů zároveň. Cílem je maximálně snížit vzájemné ovlivňování probíhajících procesů a minimalizovat tak jejich zpoždování. Zde popsaná struktura je pouze obecná, implementace jednotlivých komponentů tuto strukturu dále přizpůsobuje pro daný případ. Konkrétní softwarová realizace jednotlivých prvků systému je popsána dále v příslušných sekcích.

Program pro mikrokontroler je možné charakterizovat jako soubor procesů, které je nutné vykonávat ve stanovených intervalech, mezi tyto procesy patří obsluha snímačů, ovládání akčních členů, nebo například komunikace s PC. Tyto procesy, respektive funkce, kterými jsou implementovány, jsou volány z nekonečného while cyklu v části programu main, kde je spuštění každého z těchto procesů podmíněno spouštěcí proměnou. Tyto proměnné slouží jako rozhraní pro spuštění procesů a jsou nastavovány zpravidla pomocí přerušování. Nastavením spouštěcí proměnné je proces spuštěn ve while cyklu při nejbližší možné příležitosti. Časovače jsou nejčastěji využity pro vyvolávání přerušování, které mají funkci nastavování spouštěcích proměnných. Nastavením časovače na určitý interval je tak dosaženo vykonávání procesu s žádanou frekvencí a vhodným nastavením fáze časovače je možné minimalizovat kolize s jinými procesy o stejné spouštěcí frekvenci. Kromě časovače může být spuštění procesu podmíněno jinou událostí, například přerušováním vyvolaným po přijetí zprávy na sériové komunikaci. Kritické operace, které vyžadují rychlou odezvu, jsou v některých případech vykonávány přímo v obsluze přerušování, ale pouze v případě, že se jedná o výpočetně nenáročné úkony. Provádění výkonnostně náročných operací v obsluze přerušování by mělo za následek nahromadění přerušování a následně zpoždování ce-

5.4. SNÍMÁNÍ TLAKU

lého programu, z tohoto důvodu je minimalizována náročnost operací v obsluze přerušení a náročné procesy jsou spouštěny pomocí spouštěcích proměnných vně obsluhy přerušení.

V navrhovaném uspořádání může i tak nevyhnutelně docházet ke zpoždění spouštění procesů, například při současném požadavku spuštění všech procesů bude poslední proces zpožděn o dobu vykonání všech předchozích procesů, které jsou umístěny výše ve while cyklu. Toto je ovšem způsobeno vlastnostmi použitého mikroprocesoru, který nedovoluje paralelní vykonávání procesů ve více vláknech, avšak pomocí nástroje DMA je možné do jisté míry zajistit asynchronní vykonávání určitých procesů. Z tohoto důvodu bylo DMA využito v co nejvyšší míře pro optimalizaci programu. Příkladem využití tohoto nástroje jsou komunikační periferie, kde DMA zajišťuje „v pozadí“ přesun dat mezi pamětí a registry periférií při přijímání nebo vysílání dat a umožňuje tak procesoru vykonávat současně jiný úkon.

5.4. Snímání tlaku

5.4.1. Charakteristika snímačů tlaku

Pro účely snímání tlaku byly zvoleny snímače DMP 331i od společnosti BD Sensors (obrázek č. 5.2). Jedná se o senzor tlaku s rozsahem 0 – 10 bar, vzorkovací frekvencí 5 Hz a s digitálním výstupem RS-485 využívající protokol MODBUS RTU. Snímač disponuje mikroprocesorem, který zajišťuje mimo jiné zmíněnou komunikaci, linearizuje výstup a provádí teplotní kompenzaci měřené veličiny. Pro digitalizaci měřeného tlaku využívá snímač 16-bitový AD převodník a dosahuje přesnosti 0,1 %. Kromě tlaku je možné snímat také teplotu senzoru, v tomto případě je vzorkovací frekvence pouze přibližně 0,125 Hz, tato funkce má spíše informativní charakter. Dále, kromě aktuální hodnoty tlaku, senzor uchovává informaci o maximálním a minimálním zaznamenaném tlaku. Použité senzory jsou relativní, snímáný tlak je tedy vždy vztažen k hodnotě tlaku při zapnutí snímače, nevýhodou je pak nutnost zapnutí senzorů před natlakováním pneumatického systému.[53]

Jak bylo stanoveno v požadavcích na měření tlaku, nízká vzorkovací frekvence (5 Hz) postačuje, jelikož není vyžadováno zkoumání dynamiky změn tlaku, ale pouze statické úrovně tlaku pro udržování žádané hodnoty. Díky této nízké frekvenci může být snímání tlaku relativně výkonnostně nenáročné a bude tak mít menší dopad na zátěž mikrokontroleru.

5.4.2. EIA/TIA-485

EIA/TIA-485 běžně nazýváno také RS-485 je standard sériové komunikace, který definuje vlastnosti této průmyslové sběrnice. Sběrnice RS-485 se skládá ze dvou vodičů (běžně označovaných písmeny A,B), které přenášejí vzájemně invertovaný signál, jedná se tedy diferenciální uspořádání, které minimalizuje negativní ovlivňování komunikace vnějším



Obrázek 5.2: Snímač tlaku DMP 331i

rušením a umožňuje tak využití této sběrnice i na vzdálenosti přes 1 km, nebo s přenosovými rychlostmi do 10 Mb/s. Standard RS-485 popisuje také možnost použití 4 vodičů, neboli dvou diferenciálních párů vodičů, v tomto případě by sběrnice umožňovala sériovou komunikaci ve full-duplex módu, tato varianta je ale méně častá. Při použití jednoho páru vodičů je komunikace tedy pouze half-duplexní. Na jednu sběrnici RS-485 je možné připojit až 32 vysílačů nebo přijímačů uzlů, je tedy vhodná pro systémy s více zařízeními. RS-485 popisuje pouze elektrický standard, ovšem protokoly komunikace, fyzické provedení nebo konektory jsou na tomto standardu nezávislé. V referenčním modelu ISO/OSI se RS-485 řadí do 1. úrovně neboli do fyzické vrstvy.[54][55]

5.4.3. Hardwarová integrace

Z hlediska hardwarové integrace je nutné přizpůsobit výstup a vstup mikrokontroleru sběrnici RS-485 jelikož tuto sběrnici přímo nepodporuje. Pro tento účel byl použit převodník úrovně SN75176BP od Texas Instruments, který na hardwarové úrovni mikrokontroleru umožní komunikovat po diferenciální sběrnici jako je RS-485. Převodník je napájen napětím 5 V, kterým následně budí diferenciální linku. V případě vysílání logické jedničky mikrokontroler nastaví na vysílacím pinu napětí 3.3 V a převodník nastaví na sběrnici vodič A na 5 V a vodič B na 0 V. Naopak v případě vysílání logické nuly nastaví pře-

5.4. SNÍMÁNÍ TLAKU

vodník vodič A na 0 V a vodič B na úroveň 5 V. Pro přijímání signálů převodník úrovně porovnává napěťové úrovně vodičů A a B a v závislosti na nich nastavuje přijímací pin na 5 V v případě logické jedničky, nebo na 0 V v případě logické nuly. Mezi vysíláním a přijímáním se volí pomocí řídicího digitálního signálu.[56]

Pro komunikaci bylo využito rozhraní mikrokontroleru UART, jehož Tx a Rx piny (neboli vysílací a přijímací piny) byly připojeny na příslušné piny na převodníku. Dále byly propojeny terminály A, B převodníku a snímačů a byla tak vytvořena samotná fyzická sběrnice. Pro propojení byla zvolena sběrniceová topologie s co nejkratšími odbočkami (cca 10 cm) pro minimalizaci odrazů, ovšem vzhledem k celkově krátké délce sběrnice – přibližně 2 m, je riziko odrazů zanedbatelné.

S využitím rozhraní UART a funkcí obsažených v knihovně HAL je nyní s provedeným hardwarovým přizpůsobením možné po sériové lince RS-485 posílat data. Pro úspěšnou komunikaci se snímači je avšak nutné komunikovat v souladu s protokolem MODBUS RTU, který senzory využívají.

5.4.4. MODBUS protokol

MODBUS protokol pro sériovou komunikaci je protokol typu master-slave a v referenčním modelu ISO/OSI se řadí do druhé úrovně neboli do tzv. linkové vrstvy. Na první, fyzické, vrstvě může protokol MODBUS využívat různé standardy, jeden z nejběžnějších je však zmiňovaný standard RS-485 nebo také RS-232. Master-slave systém se skládá z jednoho zařízení typu master a jednoho nebo více zařízení typu slave, kterým master zařízení vysílá příkazy. Slave zařízení vysílají na sběrnici pouze po požadavku od master zařízení a nekomunikují s dalšími slave zařízeními. Ve zde navrhovaném systému mikrokontroler STM32F767ZI zastává roli zařízení typu master a senzory tlaku zastávají zařízení typu slave.[57]

Rámec protokolu MODBUS, který je možné vidět na obrázcích č. 5.3, začíná adresovým polem o velikosti 1 byte. V tomto poli je specifikována adresa slave zařízení, kterému je zpráva určena. MODBUS podporuje až 247 adres, adresa 0 je vyhrazena pro tzv. broadcast mód, kdy je zpráva určena pro všechny slave zařízení. V případě, kdy slave zařízení odpovídá na požadavek master zařízení, vkládá slave zařízení do adresového pole svoji adresu. Master zařízení není adresováno, přijímá všechny zprávy na sběrnici a adresové pole slouží pro identifikaci vysílajícího slave zařízení. Následující pole o velikosti 1 byte v rámci protokolu obsahuje funkční kód. Funkční kód specifikuje, jakou operaci master zařízení požaduje vykonat, typicky funkční kód definuje typ registru a zda je vyžadována operace čtení nebo zápis, existují ale také funkční kódy pro diagnostické účely. Po funkčním kódu následuje datové pole, které může nabývat velikosti 0 – 252 bytů. V případě operací zápisu a čtení je v tomto poli na začátku specifikována adresa registru ze kterého master zařízení požaduje číst nebo do kterého chce zapisovat, následuje informace o počtu bytů pro čtení, nebo zápis. V případě zápisu dále následují samotná data, která mají být slave zařízením do daného registru zapsána. V případě čtení jsou v odpovědi od slave zařízení v tomto poli obsažena požadovaná data z daného registru. Poslední pole (o veli-

kosti 2 byty) v rámci protokolu je tzv. CRC (Cyclic Redundancy Check) neboli cyklický redundantní součet. CRC slouží pro kontrolu správnosti poslaných dat a vypočítá se z celého zbytku rámce. Při přijetí zprávy je vypočítáno CRC doručených dat a následně porovnáno s CRC ze zprávy, v případě neshody obsahuje přijatá zpráva chybu.[57]

Slave Address	Function Code	Data	CRC
1 byte	1 byte	0 up to 252 byte(s)	2 bytes CRC Low ₁ CRC Hi

Obrázek 5.3: Rámec protokolu MODBUS [57]

Existují 2 typy MODBUS sériového protokolu, RTU (Remote Terminal Unit) mód a ASCII (American Standard Code for Information Interchange) mód. V RTU módu 1 znak obsahuje 1 byte dat, znakem je zde myšlena 11-bitová sekvence (obrázek č. 5.4), start bit, 8 datových bitů, paritní bit a stop bit. V RTU módu je vyžadována sudá parita, pomocí které lze detekovat chyby v jednotlivých znacích. Při posílání zprávy pomocí RTU módu může být prodleva mezi jednotlivými znaky maximálně 1,5 znaku, v případě delší prodlevy bude zpráva považována za nekompletní. Pro začátek nové zprávy musí být mezi znaky prodleva minimálně 3,5 znaku. Při použití ASCII módu je každý byte dat posílán jako dvojice ASCII znaků, na rozdíl od RTU módu je v ASCII módu začátek a konec zprávy definován předem určenými znaky a ne prodlevou mezi znaky. Z tohoto důvodu může prodleva mezi znaky v ASCII módu dosahovat až jedné sekundy, ASCII mód je proto používán u zařízení, které nesplňují časové požadavky RTU módu. V porovnání s ASCII módem je RTU efektivnější z hlediska hustoty přenesených dat, jelikož v jednom znaku přeneše 8 bitů informací. Jak bylo zmíněno, použité senzory využívají protokol MODBUS v módu RTU, v následující části bude popsána softwarová implementace tohoto protokolu.[57]

Start	1	2	3	4	5	6	7	8	Par	Stop
-------	---	---	---	---	---	---	---	---	-----	------

Obrázek 5.4: Znak módu RTU [57]

5.4.5. Softwarová integrace

Základem softwaru pro komunikaci s tlakovými snímači jsou funkce pro implementaci protokolu MODBUS RTU. Tento proces lze zjednodušeně popsat dvěma nejdůležitěj-

5.4. SNÍMÁNÍ TLAKU

šími funkcemi, funkcí pro sestavování rámců protokolu a funkcí pro zpracování přijatých rámců. Pro sestavení rámce je vyžadováno několik základních údajů: adresa slave zařízení, kterému je zpráva určena, adresa registru, funkční kód, počet bytů ke čtení/zápisu a v případě zápisu také data pro zapsání. Funkce na základě těchto údajů sestaví rámec (podle schématu na obrázku č. 5.3) a na konec rámce vloží hodnotu cyklického redundantního součtu vypočítanou ze vstupních dat funkce. Rámec je v paměti reprezentován polem proměnných datového typu „uint8_t“ neboli celočíselných 8-bitových čísel bez znaménka. Vytvořený rámec je dále poslán po sériové lince pomocí UART funkce z HAL knihovny využívající DMA – `HAL_UART_Transmit_DMA`. Díky využití DMA je proces posílání tzv. „neblokující“, další procesy programu tedy mohou probíhat paralelně s vysíláním zprávy a jejich začátek není podmíněn koncem procesu vysílání zprávy.

Druhá důležitá funkce pro implementaci protokolu je funkce zajišťující zpracování přijatých rámců. Vstupem do této funkce je pole představující obdrženou zprávu, samotný proces přijímání zprávy avšak nespadá do této funkce a bude popsán ve zbytku této podkapitoly. Před zpracováním dat obsažených v doručené zprávě je provedena řada kontrol: je ověřeno, zda byl rámec poslán žádaným slave zařízením, je ověřeno, zda rámec obsahuje správný funkční kód, je ověřeno, zda rámec dosahuje minimální délky, a nakonec je vypočítána CRC hodnota a porovnána s CRC hodnotou ze zprávy. Při úspěšném splnění těchto kontrol dále následuje zpracování samotných dat, v opačném případě je zpracování přerušeno a obsah zprávy ignorován. Zpracování dat ze zprávy spočívá v převedení dat do požadovaného datového typu. Používané snímače tlaku využívají 16-bitové AD převodníky, hodnota tlaku je tedy posílána ve formě dvou bytů v datové části rámce, po přetypování na 16-bitovou proměnnou je možné hodnotu tlaku dále předat další části programu.

Žádanou frekvenci 5 Hz snímání hodnot tlaku zajišťuje časovač. Hodnoty pro nastavení časovače na požadovanou frekvenci byly vypočítány podle vztahu č. 5.1, kde f_p je výsledná frekvence přerušování časovače, f_t je frekvence hodinového signálu, Pr je hodnota děličky a Pe je hodnota periody čítače.[58]

$$f_p = \frac{f_t}{(Pr + 1)(Pe + 1)} \quad (5.1)$$

Časovač při přetečení čítače vždy vyvolá přerušování, v tomto případě tedy každých 200 ms. V obsluze tohoto přerušování je pouze nastavena proměnná označující požadavek na odeslání patřičné MODBUS zprávy. Tato proměnná podmiňuje v hlavní while smyčce v main části programu volání funkce pro sestavení MODBUS rámce a pro odeslání zprávy. Do této funkce jsou předány přednastavené parametry adresy slave zařízení, adresy registru, funkčního kódu a počtu bytů, ze kterých je sestaven rámec a zpráva je asynchronně odeslána pomocí DMA. Od tohoto okamžiku mikrokontroler „čeká“ na odpověď snímače, na sběrnici se v této chvíli nesmí vysílat jiné požadavky. Pro tento účel je nastavena hodnota proměnné popisující stav neboli „zanepřázdněnost“ sběrnice a předchází se tak vzniku kolizí zpráv. V případě, kdy je při požadavku o odeslání zprávy sběrnice v zaneprázdněném stavu, je odeslání zprávy odloženo do uvolnění sběrnice.

Na závěr je při odeslání zprávy nutné ošetřit možnost nenávratu odpovědi, jelikož její ztráta by znamenala neomezené čekání na odpověď a z toho vyplývající zablokování sběrnice. Pro tento účel je při odeslání požadavku spuštěn časovač. Situace, kdy dojde k vyvolání přerušení tímto časovačem před obdržení odpovědi, je označena za chybný stav nedoručení odpovědi a stav sběrnice je změněn zpět na „volný“.

Přijímání zpráv na sběrnici RS-485 je podobně jako odesílání prováděno asynchronně, někdy je tato metoda označována také jako „neblokuující“ režim. Konkrétně se jedná o funkci `HAL_UART_Receive_DMA`, která pomocí DMA kontinuálně cyklicky ukládá přijatá data do stanoveného místa v paměti nezávisle na procesech vykonávaných procesorem. Využití tohoto „neblokuujícího“ režimu je stěžejní, jelikož v případě použití „blokuujícího“ způsobu čtení sběrnice by nebylo možné vykonávat žádné další instrukce do přijetí zprávy. Avšak z důvodu cyklického ukládání dat je nezbytné vytvořit způsob detekce začátku a konce přijímaných zpráv.

Protokol MODBUS RTU definuje začátek a konec zprávy prodlevou větší než 3,5 znaku mezi posílanými znaky, vhodné řešení pro detekci konce zprávy by tedy automaticky po uplynutí této prodlevy vyvolalo přerušení a upozornilo tak na ukončení doručování zprávy. Pro tento účel nabízí řada mikrokontrolerů STM32F77 „timeout“ funkci, která právě zmíněnou funkcionalitu umožňuje. Pro aktivaci této funkce jsou v inicializační části programu nastaveny bity v daných registrech a do dalšího příslušného registru je vložena žádaná hodnota prodlevy. V případě doručení zprávy je detekována prodleva 3,5 znaku a je vyvoláno přerušení, v obsluze přerušení je zkontrolován registr, který indikuje, že „timeout“ funkce detekovala prodlevu (příčin vyvolání přerušení může být více, je tedy nutné kontrolovat z jakého důvodu bylo přerušení vyvoláno). Tímto je potvrzeno, že zpráva byla doručena, stav sběrnice je možné označit jako „volný“. V tomto bodě je také vytvořena časová značka určující čas doručení dané hodnoty tlaku. Nakonec je nastavena indikační proměnná, která značí, že je třeba provést zpracování přijatého rámce, jelikož se pohybujeme stále v přerušení, není prováděno samotné zpracování přijatých dat přímo v této části programu. V části programu `main` v hlavní `while` smyčce je následovně zkontrolována hodnota této indikační proměnné a začíná proces zpracování přijatých dat. Nejprve musí být rámec extrahován z přijímacího bufferu, toto je provedeno na základě počtu přijatých bytů, který je získán s příslušného UART registru. Po izolování samotného rámce jsou tato data předána do funkce pro jejich zpracování, jejíž funkce již byla popsána.[59]

Tímto je proces čtení hodnoty tlaku ze snímače DMP 331i završen. Dále je hodnota tlaku spolu s časovou značkou předána do další části programu pro zaslání do PC.

5.5. Snímání polohy

Snímání polohy je dalším z požadavků stanovených v kapitole č. 3. Konkrétně je nutné měřit vysunutí pístu, které je spjaté s polohou testovaného objektu ve svislé ose a dále polohu kladek na lineárním vedení, na které závisí horizontální poloha testovaného objektu.

5.5. SNÍMÁNÍ POLOHY

Kromě informace o samotné poloze je dalším důvodem pro toto měření získání informace o rychlosti v jednotlivých osách.

5.5.1. Charakteristika snímačů polohy

Pro snímání polohy byl zvolen magnetický lineární enkodér TMLS-25 (obrázek č. 5.5) od společnosti JIRKA a spol., s.r.o. Jedná se o snímač s rozlišením 0,025 mm, s logickým výstupem TTL a maximální rychlostí snímání polohy 10 m/s. Snímač TMLS-25 je inkrementální a tudíž relativní, je tedy možné měřit pouze rozdíl vzdáleností od počáteční polohy. Z tohoto důvodu je nutné před začátkem měření stanovit referenční nulový bod, například začátek lineárního vedení.[60]

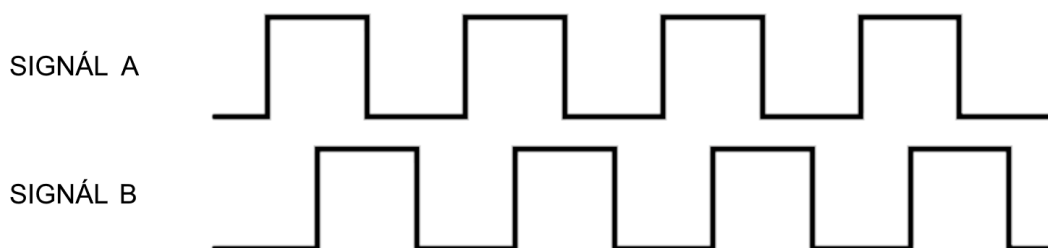


Obrázek 5.5: Lineární enkodér TMLS-25

Snímač využívá tzv. kvadrurního kódování – vytváří dva obdélníkové signály vzájemně posunuté o 90 stupňů (obrázek č. 5.6), kde každá hrana signálů představuje posunutí o 0,025 mm. Směr posunu je v kvadrurním kódování definován fázovým sledem signálů. Když stanovíme jeden směr posunu jako předbíhání druhého signálu prvním o 90 stupňů, tak směr opačného smyslu bude popsán jako druhý signál předbíhající první o 90 stupňů. Přičítáním hran při posunu v jednom směru a odčítáním hran ve směru opačném lze tímto způsobem po diskretních krocích 0,025 mm určovat změnu polohy.[61]

5.5.2. Integrace snímání polohy

Přestože by bylo poměrně jednoduché softwarově implementovat logiku zpracování kvadrurního signálu pomocí externích přerušení a jejich následným sčítáním/odčítáním, nebyl by tento přístup optimální z hlediska výpočetní náročnosti na procesor, v případě vyšší rychlosti posunu by mohlo docházet i k nahromadění přerušení, vynechání kroků a tedy i ke ztrátě informace o poloze. Výhodnější řešení tkví ve zpracování a čítání kvadrurního signálu na úrovni hardwaru, bez nutnosti využití operací procesoru. Pro tento



Obrázek 5.6: Fázově posunuté signály enkodéru [61]

účel existují specializované integrované obvody – tzv. dekodéry, avšak v případě mikrokontroleru STM32F767ZI je již tato funkcionality integrována. Konkrétně jde o speciální mód časovače, kde k inkrementaci čítače místo interního hodinového signálu slouží externí signál z enkodéru. Směr posunu je určen automaticky ze sledu signálů.[59]

Inicializace právě zmíněné funkcionality byla provedena v prostředí STM32CubeMX, kde byl příslušný časovač nastaven do módu pro dekódování kvadrurního signálu. Hodnota periody čítače v časovači byla nastavena tak, aby nemohlo dojít k přetečení čítače vzhledem k rozsahu pohybu a rozlišení snímače, dále byl zvolen mód inkrementace čítače na vzestupnou i sestupnou hranu pro oba signály. Po inicializaci časovače s tímto nastavením jsou pulzy čítány automaticky, nezávisle na dalších probíhajících procesech. Samotná informace o poloze je „skryta“ v hodnotě registru popisujícího stav čítače použitého časovače. Kdykoliv, kdy je informace o poloze v programu potřebná, je načtena hodnota registru čítače a dále je vynásobena rozlišením (v tomto případě 0,025 mm), což z tohoto přístupu činí velice efektivní a výpočetně zcela nenáročný způsob odměřování polohy.

5.5.3. Odhad rychlosti

Jelikož měřicí systém neobsahuje snímač rychlosti, je rychlost testovaného tělesa získávána nepřímou z hodnot polohy. Informace o rychlosti je určena k pozdější analýze experimentu. Diferenciace diskrétních dat polohy, pomocí které je rychlost získávána, vnáší do výsledku šum, což je nevýhoda inherentně spjatá s tímto přístupem. Z tohoto důvodu je signál v dalším kroku digitálně filtrován, nevýhodou tohoto řešení je pak zpoždování signálu. Avšak vzhledem k relativně nenáročným požadavkům na odezvu snímání rychlosti je zvoleno toto řešení oproti pokročilejším metodám estimace rychlosti.

Derivace polohy je získána rozdílem dvou po sobě jdoucích hodnot polohy, následně jsou tato data filtrována IIR filtrem neboli filtrem s nekonečnou impulzní odezvou. Konkrétně se jedná o filtr prvního stupně navržený Butterworth metodou v prostředí MATLAB, který byl implementován pomocí C kódu podle vztahu č. 5.2, kde a, b značí koeficienty

5.6. SNÍMÁNÍ ZRYCHLENÍ

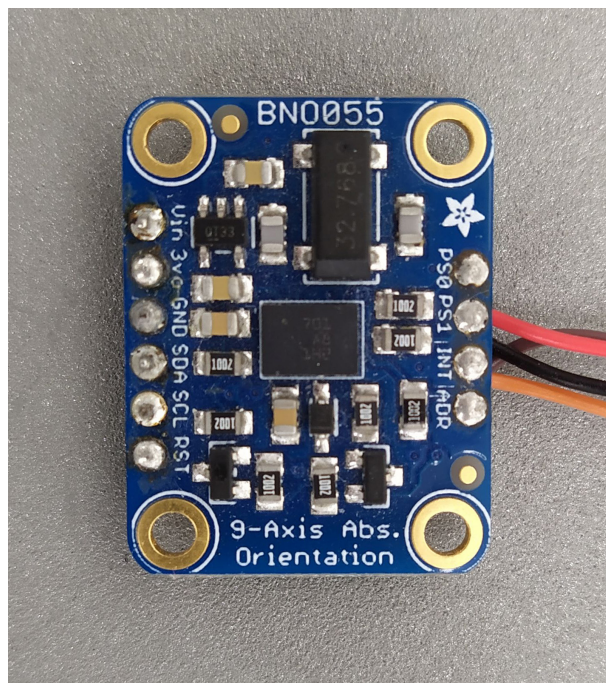
filtru, y výstupní hodnoty a x vstupní hodnoty. Filtrovaná data jsou následně posílána přes sériové rozhraní do PC.[62]

$$y_n = \frac{1}{a_1}(b_1x_1 + b_2x_{n-1} - a_2y_{n-1}) \quad (5.2)$$

5.6. Snímání zrychlení

5.6.1. Snímač

Pro snímání zrychlení na testovaném objektu byl zvolen snímač BNO055 od společnosti Bosch. Jedná se o smart snímač integrující 3 senzory – 14-bitový tříosý akcelerometr, 16-bitový tříosý gyroskop a tříosý magnetometr. Akcelerometr dosahuje maximální vzorkovací frekvence 1000 Hz a umožňuje měření zrychlení v několika rozsazích od ± 2 g až do ± 16 g. Snímač dále disponuje 32-bitovým mikrokontrolerem pro zpracování měřených dat a sériovou komunikaci (snímač podporuje rozhraní I2C a UART). Snímač nabízí kromě samotných dat ze sensorů také díky možnosti zpracování dat další veličiny, kterými jsou například relativní a absolutní natočení v prostoru vyjádřené pomocí kvaternionů, nebo Eulerových úhlů, kompas mód určující orientaci senzoru pomocí magnetometru a další. Pro účely tohoto systému bude však použit základní mód čtení „surových“ dat zrychlení, protože pouze tento mód podporuje vzorkovací frekvenci 1 kHz.[63][64]



Obrázek 5.7: Snímač BNO055

Komunikace snímače s mikrokontrolerem STM32F767ZI byla realizována pomocí rozhraní I2C, jelikož DPS (obrázek č. 5.7), na kterém byl snímač osazen, podporuje pouze tuto konfiguraci.

5.6.2. I2C

I2C je synchronní sériová half-duplexní sběrnice vyvinutá firmou Phillips. Jedná se o jednu z nejpobulárnějších sběrnic, zejména kvůli její jednoduchosti. Pro komunikaci sběrnice I2C využívá pouze 2 vodiče – vodič SCL pro hodinový signál (generovaný master zařízením) a vodič SDA pro přenos samotných dat. Ve výchozím stavu je signál na sběrnici na vyšší z napěťových úrovní, je proto vyžadováno k vodičům na sběrnici připojit pull-up rezistory. Zařízení uvádí vodiče sběrnice I2C do nižší logické úrovně jejich připojením k zemi, jedná se tedy o uspořádání typu otevřený kolektor. I2C podporuje komunikaci více zařízení na jedné sběrnici, z tohoto důvodu je nutný systém adresace zpráv. Pro tento účel slouží první byte ve zprávě, který vždy obsahuje adresu slave zařízení, kterému je zpráva určena. Adresa zařízení je reprezentována 7 bity (maximální počet zařízení na sběrnici je proto teoreticky 128), 8. bit v prvním bytu specifikuje, zda se jedná o operaci čtení, nebo zápisu do slave zařízení. V případě zápisu následují data (jeden nebo více bytů) vyslaná master zařízením, v případě čtení vysílá dále slave zařízení. Každý byte následuje tzv. „ack“ (acknowledge) bit, který slouží pro potvrzení přijetí zprávy přijímacím zařízením. Jednou z nevýhod sběrnice I2C je chybějící systém detekce chyb, není proto možné zaručit integritu přijatých dat.[65]

5.6.3. Integrace snímače zrychlení

Snímač byl připojen pomocí 4 žilového stíněného kabelu o délce 1,5 metru, kde 2 z žil slouží pro napájení a zbylé dva vodiče pro samotnou I2C komunikaci (SDA, SCL). Dále byly k oběma signálovým vodičům připojeny pull-up rezistory o hodnotě 4,7 kΩ. Jelikož mikrokontroler STM32F767ZI disponuje rozhraním I2C, je možné vodiče sběrnice připojit přímo na terminály příslušící tomuto rozhraní.

Podobně jako v případě již zmíněného snímání tlaku nebo polohy bylo pro optimální využití výpočetního výkonu mikrokontroleru implementováno měření zrychlení asynchronně. Pro čtení hodnot zrychlení z registrů snímače byla použita funkce z knihovny HAL `HAL_I2C_Master_Receive_DMA`, která jako parametry přijímá ukazatel na I2C instanci, adresu slave zařízení, ukazatel na buffer pro ukládání přijatých dat a také celočíselné číslo označující požadovaný počet bytů pro čtení. Jelikož jsou I2C adresy 7-bitové, je nutné na 8-bitovou hodnotu adresy aplikovat bitový posun doleva. Počet bytů pro čtení byl zvolen 6 – jsou čteny 3 hodnoty zrychlení, každá zabírající 2 byty (hodnoty jsou 14-bitové, 2 bity jsou tedy vždy nevyužité). Pro snížení výpočetní zátěže je dále možné snímat pouze 1 osu (svislou), v tomto případě by byl vyslán požadavek pro čtení 2 bytů. Tato funkce je spouštěna asynchronně na základě proměnné pomocí časovače s frekvencí 1000 Hz, obdobně jako v případě snímání tlaku.

5.7. SNÍMÁNÍ SÍLY

Při dokončení přijímání daného počtu bytů je vyvoláno přerušení a obsluha přerušení volá funkci `HAL_I2C_MasterRxCpltCallback`. Byla vytvořena nová definice této funkce, která specifikuje spuštění sekvence zpracování přijatých dat pomocí nastavení hodnoty indikační proměnné označující požadavek pro zpracování. Sekvence pro zpracování je dále při následujícím cyklu `while` smyčky spuštěna v části programu `main` na základě této proměnné. V rámci zpracování přijatých dat jsou binární data převedena do požadovaného datového typu a dále je těmto hodnotám přiřazena časová značka stanovující kdy byla data přijata.

Před čtením dat ze snímače je ovšem nutné provést několik operací zápisu pro nastavení snímače, konkrétně to jsou: nastavení módu snímače na hodnotu snímání pouze samotných dat z akcelerometru, nastavení rozsahu (± 16 g), frekvence (1000 Hz) a závěrem je specifikován registr pro budoucí operace čtení, tedy registr obsahující data zrychlení. Operace nastavení parametrů je provedena jednou v inicializační fázi programu, z tohoto důvodu není nezbytné tuto operaci vykonávat asynchronně.

5.7. Snímání síly

5.7.1. Hardware

Pro snímání kompenzační síly v laně byl zvolen tenzometrický snímač U9C (obrázek č. 5.8) od společnosti HBM s nominální kapacitou 200 N. Tento snímač byl zvolen především pro svoje kompaktní rozměry a nízkou hmotnost (75 g). Nízká hmotnost je zde prioritní, jelikož hmotnost snímače je přičítána k hmotnosti testovaného tělesa a dochází tak k negativnímu ovlivnění experimentu. Zesílení signálu tenzometrického snímače zajišťuje zesilovač ClipX s analogovým výstupem, taktéž od společnosti HBM.

Konverze analogové hodnoty měřené síly byla nejprve realizována interním 12-bitovým AD převodníkem mikrokontroleru, avšak od tohoto řešení bylo upuštěno z důvodu silného šumu dosahujícího až jednoho procenta měřeného rozsahu. Následně bylo zvoleno řešení využívající externí AD převodník – MCP3208-BI/P. Jedná se o 12-bitový AD převodník se sériovým rozhraním SPI a napětovým rozsahem 2,7 – 5 V, s tímto převodníkem bylo dosaženo přibližně desetkrát nižší úrovně šumu v měřeném signálu.

Pro další redukování šumu a odfiltrování vyšších frekvencí signálu byl na vstup AD převodníku přidán antialiasing filtr prvního stupně, který byl realizován pomocí RC článku. Podle požadavků na měření síly byla stanovena vzorkovací frekvence na hodnotu 1000 Hz, mezní frekvence filtru byla proto stanovena na 500 Hz, neboli na Nyquistovu frekvenci. Hodnoty RC článku byly zvoleny $R = 3183 \Omega$ a $C = 100 \text{ nF}$.



Obrázek 5.8: Snímač U9C

5.7.2. SPI komunikace

SPI – Serial Peripheral Interface neboli sériové periferní rozhraní je synchronní sériová komunikace. SPI disponuje vlastním vodičem pro oba směry toku dat, jedná se tedy o full-duplexní sběrnici. Třetí vodič rozhraní SPI slouží k přenosu hodinového signálu pro synchronizaci komunikace. Poslední vodič (tzv. SS – Slave Select, někdy také CS – Chip Select) slouží k výběru slave zařízení, se kterým master zařízení komunikuje, počet SS vodičů je roven počtu slave zařízení.[66]

Komunikace s MCP3208 má následující formát: Začátek komunikace zahajuje start bit následovaný 4-bitovou sekvencí pro konfiguraci. První konfigurační bit definuje mód (single-ended, nebo diferenciální) a následující 3 bity specifikují požadovaný kanál (MCP3208 disponuje 8 kanály). Následuje prodleva rovna přibližně 1.5 hodinového cyklu, během které je vzorkována analogová hodnota. Po této prodlevě začíná MCP3208 vysílat 12-bitovou sekvenci obsahující konvertovanou hodnotu síly. Během této transakce musí být „slave select“ signál na nižší logické úrovni a před další musí být uveden zpět do vyšší logické úrovně na dobu minimálně 500 ns.[67]

5.7.3. Softwarová integrace

Rozhraní SPI bylo inicializováno pomocí STM32CubeMX a parametry komunikace byly zvoleny podle dokumentace AD převodníku. Dále bylo inicializováno DMA pro převod dat oběma směry mezi registry periferie SPI a paměti a nakonec byl konfigurován jeden digitálně-výstupní pin pro „slave select“ signál. Přestože knihovna HAL nabízí možnost řízení signálu „slave select“, tato funkcionality zde není vhodná, jelikož HAL uvádí „slave select“ do vyšší logické úrovně po každém datovém segmentu. V tomto případě je komunikace realizována 8-bitovými segmenty a je nutné, aby „slave select“ signál setrval v nižší logické úrovni během celé transakce složené z několika 8-bitových segmentů. Z tohoto důvodu je „slave select“ řízen manuálně samostatným digitálním výstupem mikrokontroleru.

Struktura programu pro čtení hodnot z AD převodníku je analogická se snímáním zrychlení. Je využita asynchronní HAL funkce `HAL_SPI_TransmitReceive_DMA`, která je volána na základě přerušení časovače frekvencí 1 kHz. Před voláním této funkce je „slave select“ pin uveden do nižší logické úrovně. Po dokončení čtení je vyvoláno přerušení a je automaticky zavolána funkce `HAL_SPI_TxRxCpltCallback`, která oznamuje příjem požadovaného množství dat. Pomocí této funkce je „slave select“ uveden zpět do vyšší napěťové úrovně, pomocí spouštěcí proměnné je iniciován proces zpracování přijatých dat a také je v této části programu získána časová značka. Zpracování dat tkví v jejich vyjmutí z DMA bufferu a převedení na požadovaný datový typ, dále je k naměřené hodnotě přiřazena časová značka a následuje převod do náležitého formátu pro posílání po sériové lince do PC.

5.8. Ovládání motorů

Další z požadavků na řídicí jednotku je ovládání motorů pro horizontální a vertikální rozjezd. V době psaní této diplomové práce bylo prováděno testování pouze ve svislé ose, proto byl pořízen a implementován pouze motor pro vertikální rozjezd. Z tohoto důvodu zde bude popsáno ovládání tohoto motoru, motor pro vodorovný pohyb však může být přidán do systému zcela analogicky bez nutnosti dalšího vývoje.

5.8.1. Hardware

Zvolen byl krokový motor 57HS22 a driver EM806 od firmy Leadshine. Jedná se o 2-fázový krokový motor s maximálním krouticím momentem 2,2 Nm (při vhodném zapojení, tj. sériové bipolární), s 200 kroky na otáčku a se jmenovitým napětím 48 V. Driver EM806 nabízí základní způsob řízení pomocí terminálů „pulse“, „direction“ a „enable“. Při sestupné (nebo vzestupné, v závislosti na nastavení) hraně signálu na terminálu „pulse“ dochází k otočení rotoru o jeden krok (v případě mikrokrokování o příslušnou část kroku). Terminál „direction“ určuje směr otáčení, terminál „enable“ aktivuje motor. EM806 podporuje mikrokrokování od 400 kroků na otáčku až po 25600 kroků na otáčku a dále umožňuje díky sériovému rozhraní konfiguraci a ladění parametrů řízení motoru pomocí PC.

Zmíněné řídicí terminály driveru EM806 mají diferenciální provedení, je možné však realizovat ovládání i bez použití diferenciálního signálu v tzv. „single-ended“ módu. První možností je uvedení záporných terminálů na napětovou úroveň země a na kladné terminály je spínána napětová úroveň 5 V. Toto řešení není vhodné, jelikož použitý mikrokontroler pracuje s napětovou úrovní 3,3 V. Vhodnějším řešením je uvedení kladných terminálů na úroveň 5 V a spínání záporných terminálů k zemi. Toto řešení je jednoduše implementováno konfigurací digitálních výstupů mikrokontroleru v módu „open drain“.

5.8.2. Softwarová integrace

Řízení směru a aktivace motoru jsou realizovány pouhým nastavováním příslušných pinů na vyšší, nebo nižší logickou úroveň funkcí `HAL_GPIO_WritePin`. Zbývajícím signál sloužící pro polohování motoru je realizován časovačem inicializovaným v módu s PWM výstupem. Výstupní frekvence tohoto časovače odpovídá frekvenci spínání terminálu „pulse“ a je tedy úměrná rychlosti motoru. Rychlostní řízení motoru pak spočívá v možnosti regulace frekvence tohoto obdélníkového signálu.

Výstupní frekvence časovače je určena vztahem č. 5.1. Z toho vztahu je zřejmé, že změny frekvence časovače je možné docílit dvěma způsoby – změnou hodnoty děličky nebo změnou periody čítače. V tomto případě byl zvolen přístup měnění periody čítače, konkrétně je toho dosaženo zápisem požadované hodnoty do registru „ARR“ příslušného časovače. Dělička byla při inicializaci nastavena na hodnotu 0 (tj. čítač přičítá frekvencí rovnající se vstupní frekvenci 108 MHz), aby bylo dosaženo maximálního rozlišení změny frekvence. Pro příklad, při mikrokrokování nastaveném na 400 pulzů na otáčku a při žádané výstupní rychlosti 600 otáček za minutu, je požadovaná frekvence signálu 4 kHz, perioda čítače je proto nastavena na hodnotu 27 000. Při této frekvenci je nejmenší možná změna frekvence (při změně periody čítače o jedna) 0,15 Hz, což je dostatečné pro tuto aplikaci. Jelikož 600 otáček za minutu je horní hranice požadované rychlosti motoru, je změna frekvence o 0,15 Hz nejhorší případ. Pro porovnání, nejmenší možná změna frekvence při rychlosti 60 otáček za minutu je 0,0015 Hz.

Dále zbývá inicializovat šířku pulzu signálu, kterou označuje hodnota „pulse“ v konfiguraci časovače. Tato hodnota vyjadřuje, kolik kroků čítače je výstupní signál ve vyšší (případně nižší) logické úrovni, střídu je tak možné vypočítat jako poměr této hodnoty a periody čítače. Dokumentace driveru specifikuje minimální délku pulzu signálu 2,5 μ s, čítač má frekvenci 108 MHz, proto je nutné nastavit „pulse“ minimálně na hodnotu 270, v tomto případě byla nastavena hodnota 500 pro spolehlivé spínání. V případě, kdy perioda čítače klesne pod tuto hodnotu, přestane výstupní signál alternovat mezi napětovými úrovněmi – střída signálu se bude rovnat jedné, avšak pro tento případ by požadovaná výstupní frekvence musela být 216 kHz a více, což nebude při řízení motoru dosaženo.

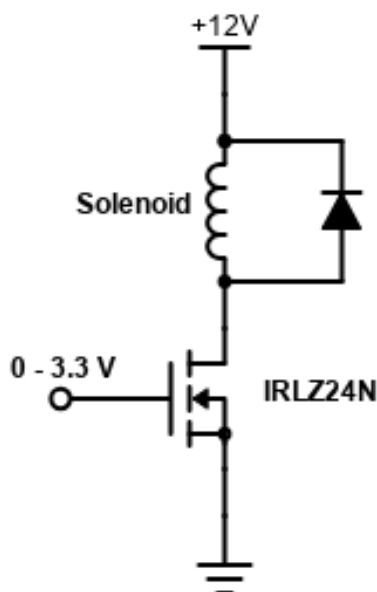
Pro ovládání motoru byla vytvořena funkce, která přijímá řídicí parametry frekvence, směru a aktivace a pomocí popsanych metod upravuje výstupní signál pro měnič krokového motoru. Voláním této funkce je tak možné dynamicky měnit výstupní frekvenci, směr, nebo je možné motor aktivovat/deaktivovat.

5.9. OVLÁDÁNÍ SOLENOIDOVÉ ZÁPADKY

Urychlení tělesa na začátku experimentu je realizováno pomocí právě zmíněné funkce postupným zvyšováním frekvence, tj. pomocí frekvenční rampy. Frekvenční rampa je definována uživatelem zadanými parametry sklonu a cílové frekvence. Sklon zvyšování frekvence odpovídá zrychlení tělesa, cílová frekvence pak cílové rychlosti. Se znalostí použité hodnoty mikrokrokování a ramena kterým motor na systém působí lze převést požadovanou rychlost na frekvenci řídicího signálu. Při urychlování tělesa je využita vlastnost krokových motorů – řízení v otevřené smyčce, z tohoto důvodu není vyžadována zpětná vazba rychlosti.

5.9. Ovládání solenoidové západky

Pro účel zajištění testovaného tělesa v nejvyšší vertikální poloze byla použita západka ovládaná pomocí solenoidu. Západka je umístěna na konci lineárního vedení tak, že v poloze maximálního vysunutí pneumatického pístu zapadá do příslušného otvoru posuvné soustavy, kterou tímto zablokuje. Ve výchozím stavu je západka vysunutá díky pružině, při aktivaci solenoidu je západka vtažena a dochází k odblokování vertikálního posuvu. Výhodou této konfigurace je bezpečnost, jelikož v krajním případě výpadku zdroje elektrické energie zůstává testovaný objekt zajištěn ve výchozí pozici.



Obrázek 5.9: Zapojení solenoidové západky

Základní parametry solenoidové cívky jsou jmenovité napětí 12 V a jmenovitý proud 400 mA. Úkolem ovládacího obvodu je připojování cívky na právě zmíněné jmenovité napětí na základě signálu z řídicí jednotky. Spínání je realizováno pomocí unipolárního MOSFET tranzistoru IRLZ24N, který svými parametry bezpečně překračuje požadované hodnoty proudu a napětí. Nízké maximální závěrné napětí tohoto tranzistoru (2 V) umožňuje jeho ovládání přímo pomocí digitálních výstupů mikrokontroleru.[68]

Schéma jednoduchého zapojení je na obrázku č. 5.9. Tranzistor je připojen sériově s ovládaným solenoidem, digitální výstup mikrokontroleru je připojen na gate terminál. Paralelně k solenoidu je připojena dioda v závěrném směru pro zabránění průrazu tranzistoru naindukovaným proudem po jeho vypnutí.

Tato konfigurace umožňuje spínání solenoidu na základě digitálního výstupu řídicí jednotky. Při výstupní hodnotě 3,3 V je solenoid aktivovaný a západka odblokuje svislý posun, při výstupní hodnotě 0 V solenoidem neteče proud a svislý pohyb testovaného tělesa je zablokován v horní poloze.

5.10. Komunikace s PC

Navázání komunikace mezi řídicí jednotkou a PC je nezbytné z několika důvodů. Předně jde o předávání naměřených hodnot, tak aby mohla být data na PC zobrazována, monitorována v reálném čase a také ukládána, jelikož řídicí jednotka nemá pro tyto účely paměťovou kapacitu. Zprávy popisující stavy řídicí jednotky a na ní probíhajících procesů jsou důležité z hlediska kontroly její funkce a pro diagnostické účely v případě poruchy. Přenos dat v opačném směru, tedy z PC do řídicí jednotky, je potřebný z důvodů konfigurace systému, například v případě změn parametrů softwaru, pro aktivaci/deaktivaci snímačů a dále také pro účely posílání příkazů ovládajících průběh experimentu.

Jednou z hlavních funkcí zařízení ST-Link V2, které je součástí vývojové desky, je vytvoření virtuálního COM portu, pomocí kterého umožňuje mikrokontroleru komunikovat s PC prostřednictvím USB rozhraní. Mezi ST-Link V2 a mikrokontrolerem STM32F767ZI je komunikace na vývojové desce realizována s využitím rozhraní UART. S využitím funkcionality nástroje ST-Link V2 tedy pro navázání komunikace stačí propojení vývojové desky (konkrétně Micro-USB příslušící zařízení ST-Link V2) a PC pomocí USB kabelu. V případě, kdyby nebylo možné ST-Link V2 využít, by pro vytvoření komunikace mezi mikrokontrolerem a PC bylo možné zvolit například externí převodník RS-232 – USB. [69]

Za pomoci prostředí STM32CubeMX bylo inicializováno rozhraní sériové komunikace USART3, které je podle schématu zapojení vývojové desky připojeno k zařízení ST-Link V2. Dále bylo inicializováno DMA pro vysílání na sběrnici a další také pro přijímání. DMA, podobně jako u ostatních periférií, umožní asynchronní implementaci komunikace.

Pro komunikaci mezi mikrokontrolerem a PC byl vytvořen následující 13-bytový formát zprávy. První byte obsahuje identifikátor, který specifikuje, o jaká data se jedná. Dalších 8 bytů slouží pro přenos samotných dat a závěrečné 4 byty obsahují sekvenci znaků „END\\n“, označující konec zprávy. Pro ilustraci, v případě vysílání naměřené hodnoty tlaku ze snímače č. 1 bude mít identifikátor hodnotu 49 (hodnota označující daný snímač), první 4 byty datové oblasti budou obsahovat 32-bitovou časovou značku, následující 2 byty budou obsahovat 16-bitovou hodnotu tlaku, 2 byty jsou nevyužity, a poslední čtyři byty obsahují koncovou sekvenci zprávy. Kromě hodnot snímačů můžou identifikátory v

5.11. NAPÁJENÍ SYSTÉMU

odchozích zprávách reprezentovat stavy určitých procesů, například pro ohlášení úspěšné inicializace nebo případně pro chybové hlášení.

Pro vysílání textových zpráv do PC byl definován další formát zprávy, který nemá pevně stanovenou délku a jednotlivé byty obsažené ve zprávě jsou interpretovány jako znaky podle ASCII. Tento formát je uvozen sekvencí znaků „msg:“, pomocí které PC detekuje, že se jedná o textovou zprávu. Tento formát zprávy umožňuje zobrazování stavových hlášek ve formě textu v grafickém uživatelském rozhraní.

Pro vysílání zpráv z mikrokontroleru byl vytvořen primární buffer (pole datového typu `uint8_t`), do kterého jsou vkládány zprávy k odeslání ve zmíněném formátu. Periodicky je ověřován stav tohoto bufferu a v případě přítomnosti nových položek jsou tato nová data k odeslání přesunuta do DMA bufferu. Následně je volána funkce `HAL_UART_Transmit_DMA`, která začne asynchronně obsah DMA bufferu vysílat. Byla zvolena konfigurace s dvěma buffery, aby nedocházelo k modifikaci obsahu DMA bufferu během vysílání. Tímto způsobem mohou být data k odeslání vkládána do primárního bufferu i během samotného procesu vysílání.

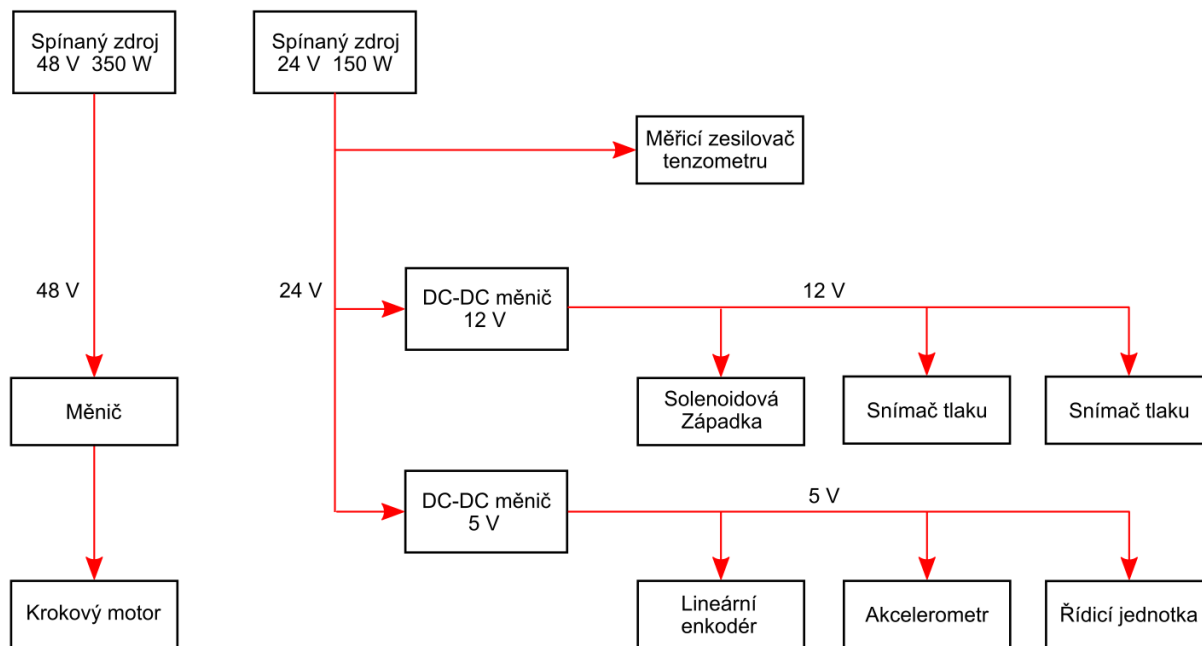
V případě přijímání dat mikrokontrolerem byl využit cyklický mód DMA, při kterém je přijímací funkce `HAL_UART_Receive_DMA` volána pouze jednou při inicializaci, dále automaticky asynchronně ukládá příchozí data do určeného bufferu. Stejně, jako v případě komunikace se snímači tlaku, jsou registry periferie USART3 konfigurovány tak, aby došlo k vyvolání přerušení po určité prodlevě na sběrnici, konkrétně byla zvolena doba prodlevy odpovídající době vyslání 20 bitů. Toto přerušení upozorní na přítomnost nově přijatých dat pomocí nastavení příslušné indikační proměnné. Z části programu `main` je dále v závislosti na této proměnné volána funkce pro zpracování přijatých dat. Z registru periferie je zjištěno kolik bytů bylo přijato (na základě rozdílu poslední pozice čtení a zápisu), tyto data jsou zkopírována z DMA bufferu a vložena do funkce pro identifikaci jednotlivých zpráv z pole přijatých bytů. Zprávy jsou identifikovány v poli detekcí koncové 4bytové sekvence „END\\n“ a následně jsou ze zprávy extrahovány data, na základě kterých jsou vykonány příslušné úkony. Například příchozí zpráva s identifikátorem 49 (označující snímač tlaku č. 1), podle hodnot v datové části může daný snímač aktivovat nebo deaktivovat. Identifikátory ve zprávách nejsou vázány pouze ke snímačům, můžou také označovat například softwarová nastavení aj.

5.11. Napájení systému

Elektronický subsystém obsahuje řadu komponentů, každý s individuálními napájecími požadavky, úkolem napájecího systému je zajistit každému zařízení napájení s vhodným napětím a proudem. Mnohé z komponentů disponují možností napájení napětím v určitém rozsahu (namísto nutnosti napájení diskrétní napětovou úrovní), lze proto v závislosti na napětových rozsazích zařízení vybrat takové napětové úrovně, aby byl jejich počet minimalizován. V případě tohoto systému byly vybrány 3 napětové úrovně: 5 V, 12 V a 24 V, které dohromady splňují požadavky všech zařízení. Zde uvažovaná zařízení nezahrnují

elektromotory, které jsou napájeny odděleně individuálními zdroji, nezávisle na napájení ostatní elektroniky.

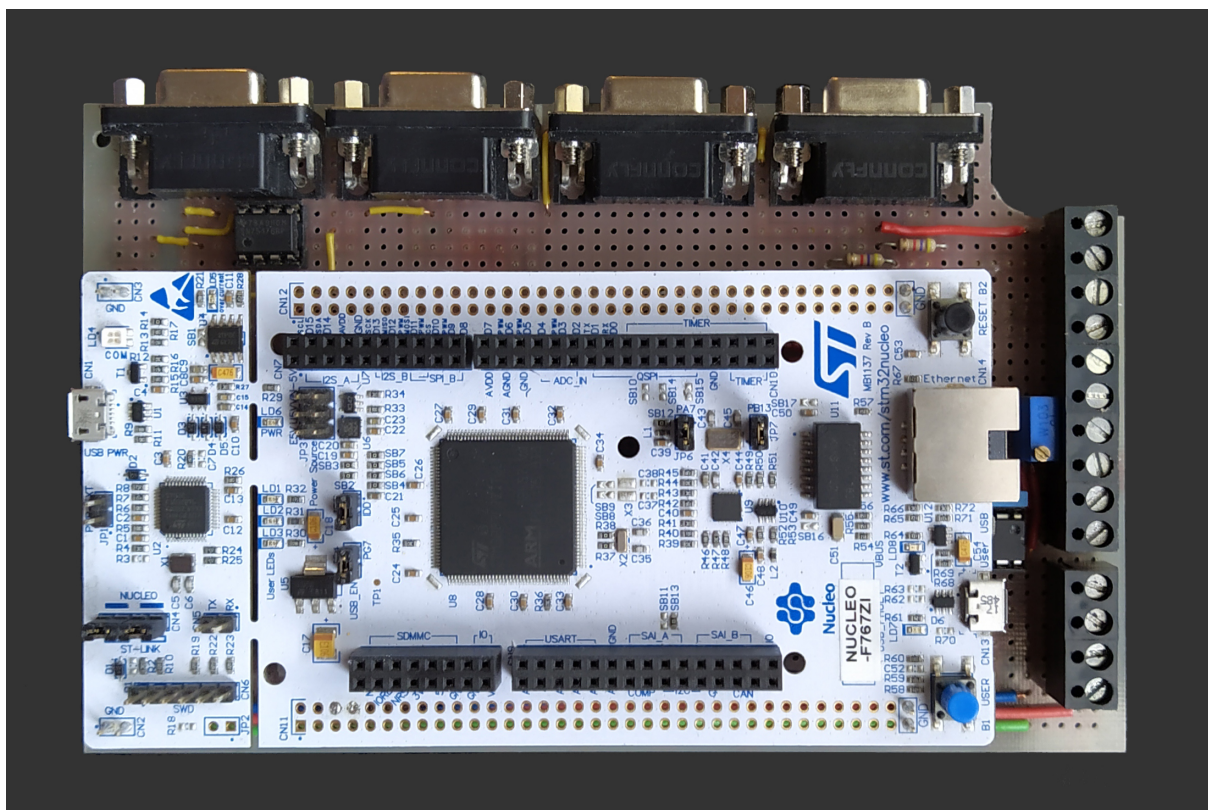
Pro vytvoření napěťových úrovní 5 V, 12 V, 24 V, byla zvolena konfigurace skládající se z 24 V spínaného zdroje Mean Well LRS-150-24 o výkonu 150 W a dvou DC-DC snižujících měničů pro napětí 5 a 10 V. Diagram zapojení napájecího systému je na obrázku č. 5.10.



Obrázek 5.10: Schéma napájení elektronického subsystému

5.12. Realizace hardwaru

Během vývoje řídicí jednotky byla popsána hardwarová řešení realizována pomocí desky univerzálního tištěného spoje. Účelem této desky je zajištění rozhraní mezi periferiemi a vývojovou deskou Nucleo-F767ZI a sjednocení vstupů a výstupů do jednoho zařízení představujícího řídicí jednotku. Zmíněnou desku je možné vidět na obrázku č. 5.11. Vstupy a výstupy této desky zahrnují svorky pro napájení napětími 5 a 12 V, konektory DE-9 pro připojení senzorů tlaku, enkodérů a akcelerometru, svorky pro snímání analogového signálu síly, svorky pro ovládání motoru a solenoidové západky a také patiči pro zapojení vývojové desky. Po dokončení vývoje a otestování hardwaru bude možné realizovat řídicí jednotku pomocí řádného tištěného spoje obsahujícího samotný mikrokontroler STM32F767ZI s požadovanými periferiemi.



Obrázek 5.11: Řídící jednotka

6. Návrh grafického uživatelského rozhraní

Grafické uživatelské rozhraní na PC má za úkol zprostředkovat interakci uživatele s testovacím systémem, konkrétně uživateli umožní v reálném čase monitorovat měřené veličiny, stavy elektronického subsystému a také ovládat elektronický subsystém. Dále bude grafické uživatelské rozhraní mimo jiné umožňovat ukládání dat pro pozdější analýzu. Pro splnění právě zmíněných požadavků bude vytvořen program, který bude zpracovávat přichodící data a vysílat data s využitím virtuálního COM portu, bude vytvářet grafické rozhraní obsahující grafy veličin obnovované v reálném čase, tlačítka, přepínače a další prvky pro interakci s řídicí jednotkou a její konfiguraci. Program bude také disponovat funkcemi pro práci se soubory, za účelem ukládání dat. C++ bylo zvoleno jako programovací jazyk pro naprogramování této aplikace ve spojení s Qt pro vytvoření samotného grafického rozhraní. Aplikace uživatelského rozhraní je vyvíjena pro operační systémy Windows.

6.1. Qt

Qt je multiplatformní framework pro vývoj aplikací se zaměřením na vytváření grafických rozhraní. Podporované platformy zahrnují Linux, OS X, Windows, Android, iOS, BlackBerry a další.[70]

Qt je možné považovat za rozšíření C++, které pomocí knihoven Qt a preprocesoru Meta-Object Compiler (MOC) přidává C++ další funkcionalitu zejména pro vytváření GUI, ale také v mnoha dalších oblastech, jako například pro práci s multimédií, sítěmi, vlákny, databázemi SQL, periferiemi jako Bluetooth, nebo NFC aj. Mezi další funkcionalitu přidanou díky preprocesoru MOC se řadí například systém signálů a slotů, které slouží pro provázání Qt objektů v rámci grafického rozhraní.[71]

Qt poskytuje také specializované vývojové prostředí Qt Creator pro vývoj aplikací založených na Qt. Qt Creator disponuje textovým editorem pro psaní kódu s kontrolou syntaxe, grafickým editorem pro vytváření Qt grafických rozhraní, nástrojem pro ladění aplikací, podporou verzovacích systému a dalším. Prostředí Qt Creator bylo využito pro vývoj aplikace uživatelského rozhraní.[72]

Pro Qt framework je vyvíjena řada volně dostupných knihoven a widgetů pro jeho další rozšíření. Jednou takovou knihovnou je QCustomPlot – volně dostupná knihovna pro vytváření grafů a vizualizací dat v rámci Qt grafických rozhraní. Grafy knihovny QCustomPlot jsou vysoce přizpůsobitelné, výkonově nenáročné a vhodné pro aplikace vyžadující zobrazování a obnovování dat v reálném čase. Nástroj QCustomPlot byl proto dále využit pro grafické zobrazování dat ve vyvíjeném grafickém uživatelském rozhraní.[73]

6.2. Struktura programu uživatelského rozhraní

Navrhovaný program je možné rozdělit do dvou hlavních procesů. Prvním je proces komunikace, do kterého spadá kontinuální přijímání dat, implementace komunikačního protokolu a základní zpracování dat. Druhým procesem je proces spravující grafické rozhraní, který řídí jeho obnovování, vykreslování přijatých dat do grafů a jeho součástí jsou také části programu reagující na akce uživatele. Je důležité zamezit vzájemnému zpoždování těchto dvou procesů, aby nedocházelo například ke ztrátám při přijímání dat, nebo k prodávám při obnovování grafického rozhraní. Z tohoto důvodu byly tyto dva procesy přiřazeny do samostatných vláken (tzv. multithreading), je tím tak dosaženo nezávislosti obou procesů. K rozdělení procesů do samostatných vláken byla použita třída QThread, konkrétně je vytvořeno nové vlákno typu QThread, do kterého je následně přiřazena instance třídy obsahující proces přijímání dat pomocí funkce `moveToThread`. Pro interakci s objektem v novém vlákne bylo vytvořeno propojení systémem signálů a slotů. Vyvoláním příslušného signálu je poté spuštěn samotný proces přijímání dat, který bude vykonáván v samostatném vlákne. Zbytek programu probíhá nezávisle dále v původním vlákne.

6.2.1. Sériová komunikace

Pro účel přijímání a vysílání dat pomocí virtuálního COM portu, byla vytvořena třída využívající funkce knihovny windows.h (Windows API funkce). Tato třída sériové komunikace obsahuje funkci pro inicializaci sériové komunikace, funkci pro přijímání dat a funkci pro vysílání. Inicializace sériového rozhraní využívá funkci `CreateFileA`, která vytvoří referenci na příslušný COM port a dále je k sériovému rozhraní přistupováno jako k souboru – zápisem do tohoto „souboru“ jsou data vysílána přes sériové rozhraní, čtením jsou data ze sériové linky přijímána. Analogicky jsou tedy ve funkcích pro vysílání a přijímání dat využity funkce `readFile` a `writeFile` knihovny windows.h. Tato implementace sériové komunikace je blokující, což znamená, že operace čtení zabrání dalšímu postupu, dokud není přijato požadované množství dat. Jelikož se ale proces přijímání dat nachází v samostatném vlákne, bude pozastaveno pouze dotyčné vlákno, ostatní procesy nebudou ovlivněny.

Samotný proces přijímání dat má podobu nekonečného cyklu while. Nejprve je v cyklu volána funkce pro příjem dat, data jsou přijímána po jednotlivých bytech. Přijatý byte je vložen do přijímacího vektoru a dále je kontrolován počet přijatých znaků ve vektoru. V případě, kdy počet dosáhne 13 bytů, tj. velikosti zprávy používaného formátu a zároveň souhlasí koncová sekvence zprávy „END\\n“, přijímání zprávy je dokončeno a lze obsah zprávy zpracovat. První byte zprávy (identifikátor), specifikuje, o jaká data se jedná. První čtyři byty datového pole obsahují časovou značku, samotná hodnota se pak nachází v druhé polovině datového pole. Tyto data jsou vyjmuta, převedena na příslušné datové typy a následně vložena do vektorů pro jejich uchování. Tyto vektory jsou globální, čímž je umožněno jejich předání části programu spravující grafické rozhraní.

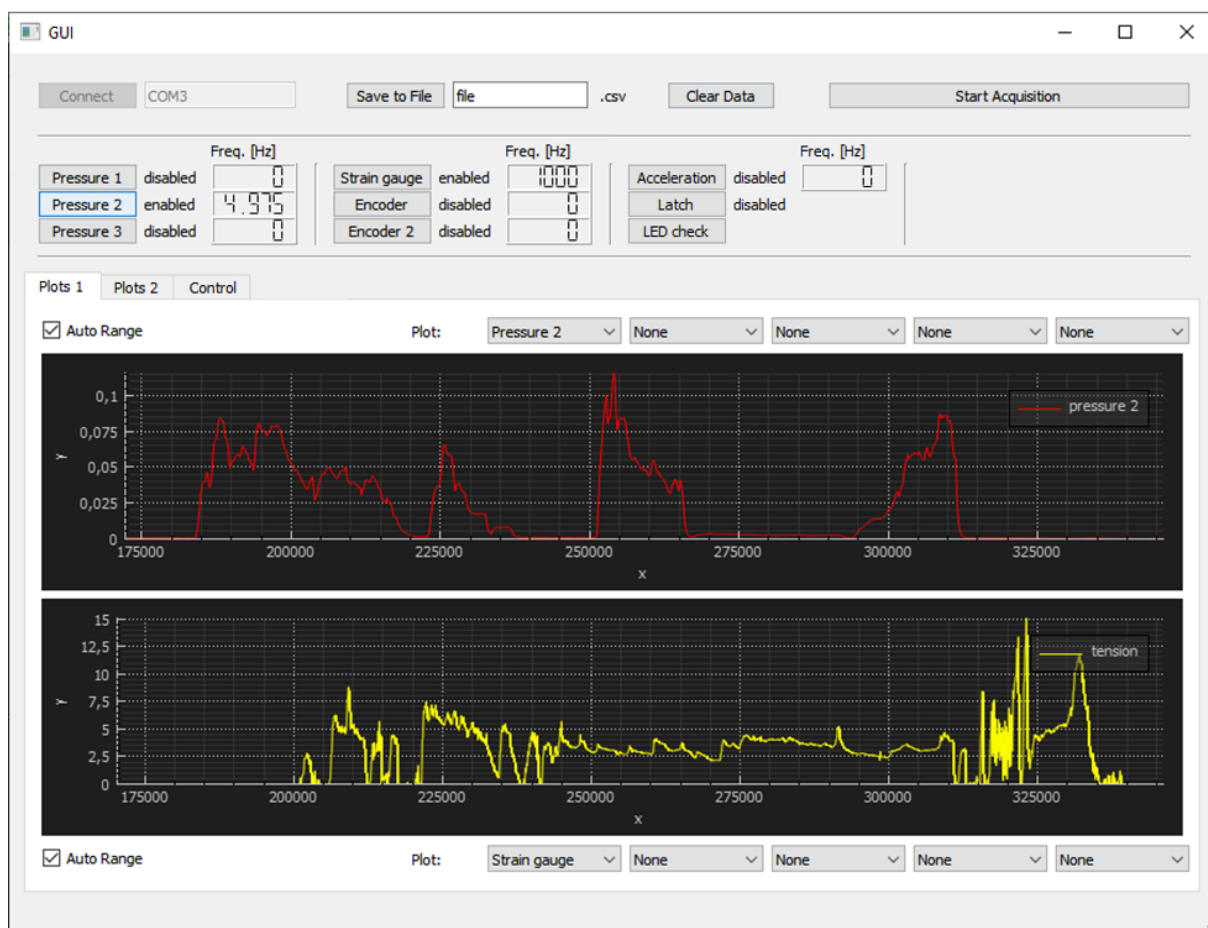
6.2.2. Grafické rozhraní

Aplikace byla vyvíjena ve zmíněném vývojovém prostředí Qt Creator, které nabízí grafický editor pro navržení Qt grafického uživatelského rozhraní. Qt Creator mimo jiné dále umožňuje automatickou inicializaci kódu tzv. slotů – „funkcí“, které jsou volány při interakci s grafickým rozhraním (přesněji, sloty jsou spouštěny v reakci na signály, které jsou vysílány objekty v grafickém rozhraní). Do slotů je následně doplněn kód, který je vykonán při určité akci uživatele. Příkladem je proces inicializace komunikace pomocí stisku tlačítka a textového pole. Do textového pole je vepsáno označení COM portu, ke kterému se uživatel chce připojit (například „COM3“) a stiskem tlačítka je vyslán signál pro zahájení připojení. Kód obsažený ve slotu příslušící akci stisku tohoto tlačítka je spuštěn, načte hodnotu z textového pole a předá ji inicializační funkci sériové komunikace. Funkcionalita ostatních ovládacích prvků v grafickém rozhraní je zajištěna analogicky.

Pro obnovování grafického rozhraní byla vytvořena nekonečná while smyčka, která v každém cyklu volá funkce zajišťující aktualizaci jednotlivých prvků grafického rozhraní. Jedním z prvků vyžadující kontinuální obnovování je grafické zobrazování přijatých dat v grafech. Po detekci nových dat v globálních vektorech jsou tato data předána k vykreslení

do grafů pomocí funkcí knihovny QCustomPlot. Dalšími z prvků, které vyžadují kontinuální obnovování, je zobrazení stavu snímačů, jejich aktuální vzorkovací frekvence a nebo také terminál, který zobrazuje přijaté stavové zprávy.

6.3. Grafický návrh a funkcionality

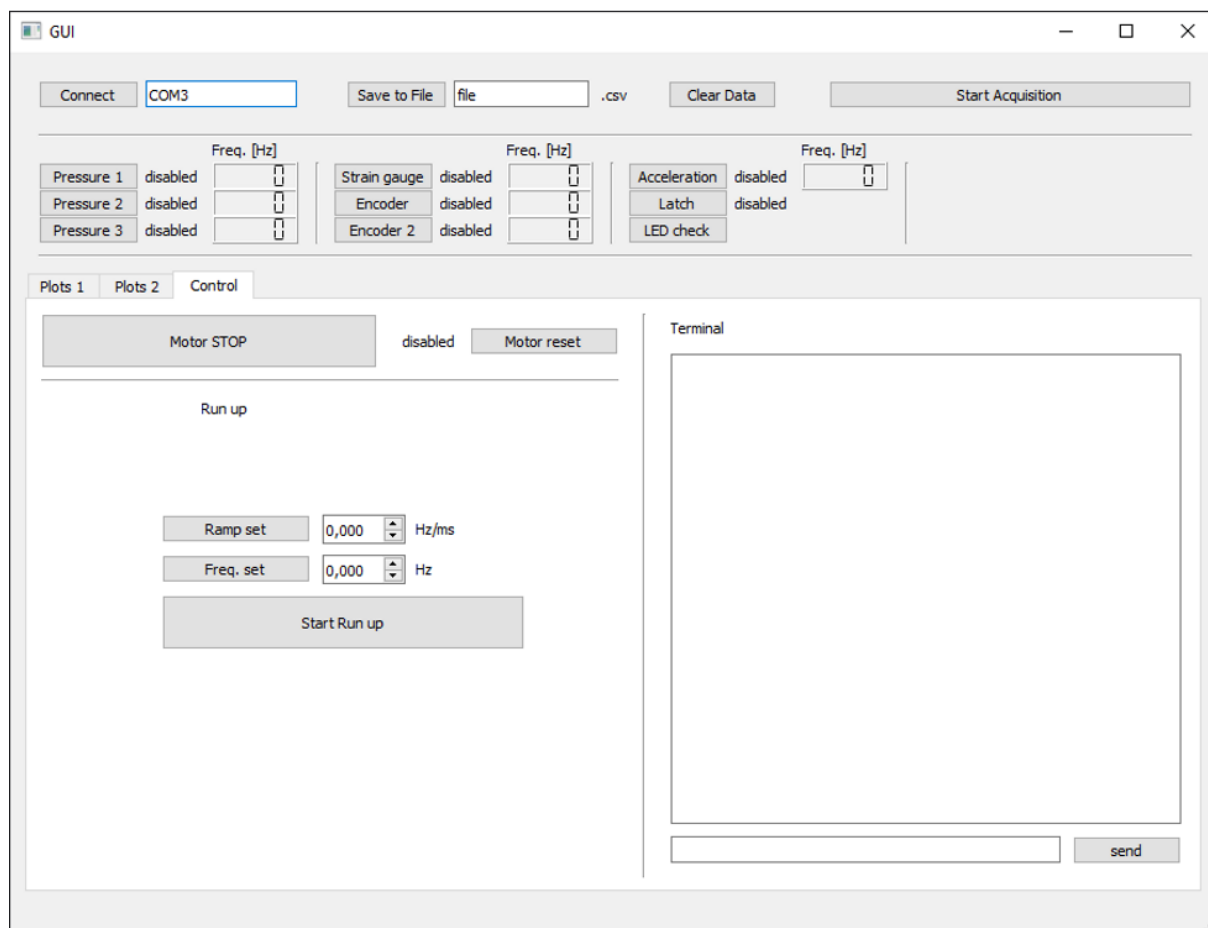


Obrázek 6.1: Grafické uživatelské rozhraní

Vzhled grafického uživatelského rozhraní je možné vidět na obrázku č. 6.1. V levé horní části okna uživatelského rozhraní se nachází prvky pro inicializaci komunikace mezi PC a mikrokontrolerem, konkrétně se jedná o textové pole pro specifikaci COM portu a tlačítko pro připojení. Informace o úspěšnosti, nebo případné neúspěšnosti připojení je uživateli zobrazena ve vyskakovacím okně. Další akce v uživatelském rozhraní jsou podmíněny úspěšným navázáním spojení s řídicí jednotkou. V řadě následuje rozhraní pro ukládání naměřených dat, pomocí kterého je možné po specifikaci názvu souboru (nebo cesty k souboru) uložit všechna naměřená data. Data jsou ukládána do souboru typu csv ve formátu časových řad. Vpravo od rozhraní ukládání se nachází tlačítko pro vymazání

6.3. GRAFICKÝ NÁVRH A FUNKCIONALITA

naměřených dat v programu, tato akce ovšem neovlivňuje uložená data v souborech. Posledním prvkem v horní části programu je tlačítko pro zahájení akvizice dat, jeho stisknutím dojde k započetí ukládání příchozích dat. Pod horizontální dělicí čarou je umístěno rozhraní pro aktivaci a deaktivaci komponentů systému. Stiskem příslušného tlačítka je řídicí jednotce vyslán příkaz k aktivaci komponentu (případně k deaktivaci pokud je již prvek aktivní). Stav daného komponentu je zobrazen napravo od tlačítka a dále, v případě snímačů, je zobrazena také jejich aktuální vzorkovací frekvence, která je vypočítávána z časových údajů příchozích dat.



Obrázek 6.2: Grafické uživatelské rozhraní, karta ovládání experimentu

Pod další horizontální čarou se nachází widget s kartami. První dvě karty obsahují každá 2 grafy pro zobrazení časové závislosti dat, které umožňují libovolné upravení rozsahů, včetně automatického přizpůsobení rozsahů a také zobrazení až 5 časových řad v jednom grafu. V poslední kartě (obrázek č. 6.2) se nachází rozhraní pro řízení experimentu. Tato karta obsahuje v horní části tlačítko pro nouzové zastavení motoru a tlačítko pro reset motoru, který je nutný v případě přetížení motoru. Dále toto rozhraní obsahuje prvky pro nastavení parametrů rozjezdu, konkrétně se jedná o sklon frekvenční rampy (v jednotkách Hz/s) a cílovou frekvenci, která má být dosažena. Oba tyto parametry se vztahují k signálu pro řízení krokového motoru a v závislosti na použitém mikrokrokování a délce

6. NÁVRH GRAFICKÉHO UŽIVATELSKÉHO ROZHRAŇÍ

ramena motoru je možné si frekvenci přepočítat na výslednou rychlost. Při nastavování parametrů rozjezdu je důležité brát v potaz omezenou délku posuvu a moment motoru. Pod rozhraním pro nastavení parametrů se nachází tlačítko pro spuštění rozběhu, a tedy i pro zahájení experimentu. Posledním prvkem v kartě ovládání experimentu je komunikační terminál, který zajišťuje vypisování stavových hlášek systému a dále také uživateli umožňuje manuální posílání příkazů řídicí jednotce.

7. Řízení a průběh experimentu

Prvním krokem při přípravě před testováním je zapnutí napájení elektronického subsystému, tímto je dosaženo inicializace snímačů tlaku, které přijmou aktuální úroveň tlaku (při nenatlakovaném systému) jako svoji referenční nulovou tlakovou hladinu. V dalším kroku může být navázáno spojení mezi řídicí jednotkou a uživatelským rozhraním na PC specifikací příslušného COM portu a stiskem tlačítka „connect“ v grafickém rozhraní. Dialogové okno informuje uživatele o úspěšnosti navázání spojení. Následně je provedena aktivace snímačů. Po stisku tlačítka snímače se v případě úspěšné aktivace daného senzoru změni stav přilehlého stavového indikátoru a zároveň se zobrazí vzorkovací frekvence dat z tohoto snímače. Vybraná snímaná data je možné zobrazit jejich výběrem v rozbalovacím seznamu v kartě grafů. Díky zobrazení snímaných hodnot tlaku je dále možné natlačit systém na předem stanovenou hodnotu a dosáhnout tak požadované síly. Po natlakování systému je testované těleso umístěno do počáteční pozice, tj., nejvyšší polohy svislého posunu. V této poloze je stiskem příslušného tlačítka v horní části uživatelského rozhraní aktivována solenoidová západka, která do započetí samotného experimentu břemeno v počáteční poloze zajistí. Následuje nastavení hodnot rozjezdové rychlosti a rozjezdového zrychlení v kartě ovládání experimentu, nastavené hodnoty je možné zkontrolovat v kartě terminálu, kde jsou po provedení změn řídicí jednotkou hodnoty vypsány. Stiskem tlačítka „Start“ v kartě ovládání experimentu je experiment započat, nejdříve je deaktivována solenoidová západka a následně začíná motor urychlovat těleso stanoveným zrychlením, dokud není dosažena cílová rychlost. V momentě dosažení požadované rychlosti je motor mechanicky odpojen a následuje dopad, během kterého elektronický subsystém provádí měření. Po dokončení testu je možné deaktivovat snímače (pro zabránění dalšímu měření dat) a díky grafickému zobrazení dat lze měření analyzovat již v uživatelském rozhraní. Pro další analýzu a sdílení dat lze data exportovat rozhraním v horní části okna, které umožní uživateli data uložit ve formátu csv.

8. Zhodnocení

První funkcí integrovanou do systému byla funkce snímání tlaku. Použité relativní snímače dosahují vysoké přesnosti, ale nízké vzorkovací frekvence – pouze 5 Hz. Rychlé dynamické změny tlaku proto nemohou být detekovány, snímače umožňují sledovat pouze dlouhodobější vývoj tlaku. Díky použité průmyslové sběrnici RS-485 je snímání tlaku robustní a málo náchylné k rušení. Sběrnice dále umožňuje připojení dalších senzorů tlaku bez nutnosti modifikace hardwaru řídicí jednotky. Nevýhodou je relativita snímačů, kvůli které je nutné snímače inicializovat při nenatlakovaném pneumatickém systému.

Snímání polohy bylo realizováno pomocí magnetického lineárního enkodéru. Díky speciální funkci mikrokontroleru využívající časovače pro dekodování dat z enkodéru je snímání polohy minimálně náročné na výkon a velice stabilní. Nevýhodou při měření polohy je průtah lana, který způsobuje chybu mezi reálnou polohou tělesa a polohou vozíku na lineárním vedení, která je měřena enkodérem. Pro preciznější stanovení polohy tělesa by bylo nutné použít například vizuální metody.

Pro měření zrychlení na zkoumaném tělese byl použit snímač BNO055. Tento snímač dosahuje maximální vzorkovací frekvence zrychlení 1000 Hz, což splňuje stanovené požadavky, ale oproti snímání tlaku má měření zrychlení kvůli frekvenci 1000 Hz podstatně vyšší výpočetní náročnost. Nevýhodou tohoto řešení je dále kabel vedoucí k testovanému tělesu, který by teoreticky mohl mechanicky ovlivňovat experiment, snímání zrychlení by tedy mohlo být dále vylepšeno o bezdrátovou komunikaci.

Snímání kompenzační síly zařízení je prováděno pomocí tenzometru. Signál z tenzometru je zesílen zesilovačem, analogová hodnota ze zesilovače je dále digitalizována externím AD převodníkem, se kterým komunikuje mikrokontroler pomocí rozhraní SPI. Minimální požadovaná vzorkovací frekvence 1000 Hz byla dosažena, avšak oproti ostatním snímačům je vzorkovací frekvenci snímání síly možné dále navýšit. Podobně jako u akcelerometru by kabel vedoucí k tenzometru mohl ovlivnit měření, ovšem u potenciálního zařízení s větší kapacitou by toto riziko (vzhledem k vyšší hmotnosti testovaného tělesa) bylo zanedbatelné.

V požadavcích na elektronický subsystém stanovených v úvodní části práce byly požadovány 2 motory – jeden pro rozjezd ve vertikální a jeden pro rozjezd v horizontální ose posuvu. V době psaní této práce byl vyžadován pro testování jen motor urychlující těleso vertikálně a byl proto tedy pořízen pouze jeden motor. Ze stejného důvodu byl implementován lineární enkodér pouze pro vertikální posun. Druhý motor a druhý lineární enkodér pro horizontální posun je ovšem možné do systému přidat analogicky bez dodatečného vývoje, jelikož řídicí jednotka je na přidání těchto komponentů připravena. Použití krokového motoru má výhodu jednoduchého ovládání. Další výhodou je možnost řízení v otevřené smyčce, krokový motor tak umožňuje provést urychlení tělesa bez zpětné vazby rychlosti.

Pomocí C++ a Qt bylo úspěšně vytvořeno grafické uživatelské rozhraní pro PC, které umožňuje jednoduché a intuitivní ovládání systému. Jeho nevýhodou je kompatibilita

pouze s operačními systémy Windows z důvodu využití Windows API pro účely komunikace s řídicí jednotkou.

9. Závěr

Diplomová práce se věnuje zařízení sloužícímu pro simulaci dopadů za podmínek snížené gravitace. Jedná se o komplexní systém, který byl rozdělen na několik subsystémů, těmi jsou například mechanický, pneumatický, nebo elektronický subsystém. Cílem této práce byl návrh elektronického subsystému na základě stanovených požadavků.

V úvodní části práce byly popsány metody používané pro simulaci snížené gravitace, byly to jak metody využívající volný pád, tak i metody založené na principu kompenzace tíhové síly, do kterých se řadí i zařízení z této diplomové práce. Kapitola „Možnosti řízení a sběru dat v podobných aplikacích“ pojednávala o možnostech provedení navrhovaného systému, konkrétně se porovnávaly možnosti: řízení pomocí PLC, řízení pomocí PC, a nakonec řízení pomocí vestavěného systému vytvořeného na základě mikrokontroleru. V úvodní části diplomové práce byl také představen projekt Zero-G, na kterém se tato práce podílí a byly vytyčeny požadavky pro navrhovaný subsystém. Na základě těchto požadavků byly dále zvoleny jednotlivé komponenty systému. Jako typ řízení bylo zvoleno řízení na bázi vestavěného systému, tento přístup řízení a snímání dat snížil náklady na komponenty systému a zvýšil flexibilitu systému, jelikož řídicí jednotka byla s pomocí mikrokontroleru vytvořena „na míru“. Nevýhodou tohoto přístupu byl ovšem delší a náročnější vývoj.

V druhé, praktické, části diplomové práce byl popsán návrh a realizace subsystému, který se skládal především z úloh snímání tlaku, polohy, zrychlení a síly a z procesu řízení krokového motoru. Integrace těchto jednotlivých prvků systému byla popsána jak z hlediska hardwaru, tak i softwaru. Jelikož navrhovaný systém obsahuje mnoho komponentů, a tudíž i mnoho procesů mikrokontroleru, byl při vytváření softwaru pro mikrokontroler kladen důraz na co nejvyšší efektivitu programu, tak aby byly dosaženy žádané frekvence pro snímání dat a řízení. Tento cíl byl splněn. Kromě softwaru pro mikrokontroler, bylo dále naprogramováno i uživatelské grafické rozhraní pro PC pro účely interakce se systémem.

Závěrem, elektronický subsystém byl úspěšně navržen, realizován a otestován a požadavky vytyčené v úvodu diplomové práce byly splněny. Elektronický subsystém plní svoji stanovenou funkci, je zde ovšem potenciál systém dále modifikovat a rozšiřovat o další funkcionalitu. Jednou z funkcí, o kterou by mohl být systém v budoucnu rozšířen, je například aktivní kompenzace tření, která by mohla využívat funkci snímání síly a ovládání motoru, které jsou již implementovány. Jako celek, testovací zařízení „Zero-G“ pomohlo k eliminaci mnoha neznámých na základě testování a optimalizace komponent ve zmenšeném měřítku. V budoucnu bude systém obohacen a jeho nová verze bude použita k dopadovému testování plné velikosti.

Literatura

- [1] How Do We Know It's Really Microgravity? NASA [online]. Washington, D.C.: NASA, 2013 [cit. 2020-06-09]. Dostupné z: https://www.nasa.gov/mission_pages/station/research/news/microgravity.html
- [2] HARLAND, David M. Microgravity. Encyclopædia Britannica [online]. Chicago: Encyclopædia Britannica, 2012 [cit. 2020-06-09]. Dostupné z: <https://www.britannica.com/science/microgravity>
- [3] Microgravity Research Program. NASA [online]. Washington, D.C.: NASA, 1998 [cit. 2020-06-09]. Dostupné z: <https://www.nasa.gov/centers/marshall/news/background/facts/microgravity.html>
- [4] Parabolic Flight. NASA [online]. Washington, D.C.: NASA, 2019 [cit. 2020-06-09]. Dostupné z: <https://www.nasa.gov/analogs/parabolic-flight>
- [5] NASA Sounding Rocket Program Overview. NASA Sounding Rocket Program [online]. Washington, D.C.: NASA, c2020 [cit. 2020-06-09]. Dostupné z: <https://rscience.gsfc.nasa.gov/srrov.html>
- [6] Drop towers. European Space Agency [online]. Paříž: European Space Agency, 2013 [cit. 2020-06-09]. Dostupné z: https://www.esa.int/Science_Exploration/Human_and_Robotic_Exploration/Research/Drop_towers
- [7] Philae. NASA [online]. Washington, D.C.: NASA, 2004 [cit. 2020-06-09]. Dostupné z: <https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=2004-006C>
- [8] Rosetta. NASA [online]. Washington, D.C.: NASA, 2004 [cit. 2020-06-09]. Dostupné z: <https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=2004-006A>
- [9] Scientific research flights. Air Zero G [online]. Paříž: Air Zero G, c2019 [cit. 2020-06-09]. Dostupné z: <https://www.airzerog.com/scientific-research-services/>
- [10] Microgravity and ISS. European Space Agency [online]. Paříž: European Space Agency, 2018 [cit. 2020-06-09]. Dostupné z: https://www.esa.int/Education/Orbit_Your_Thesis/Microgravity_and_ISS2
- [11] Falling upwards: how to create microgravity. European Space Agency [online]. Paříž: European Space Agency, 2018 [cit. 2020-06-09]. Dostupné z: https://www.esa.int/Science_Exploration/Human_and_Robotic_Exploration/Research/Falling_upwards_how_to_create_microgravity
- [12] Microgravity and drop towers. European Space Agency [online]. Paříž: European Space Agency, 2018 [cit. 2020-06-09]. Dostupné z: http://www.esa.int/Education/Microgravity_and_drop_towers

- [13] JAGANNATHAN, S., R.C. FENN a B.G. JOHNSON. Low-cost active anti-gravity suspension system. In: Proceedings of 1995 American Control Conference - ACC'95 [online]. Seattle: American Autom Control Council, 1995, s. 3606-3611 [cit. 2020-06-09]. DOI: 10.1109/ACC.1995.533809. ISBN 0-7803-2445-5. Dostupné z: <http://ieeexplore.ieee.org/document/533809/>
- [14] LEIPOLD, M, M EIDEN, C.E GARNER, et al. Solar sail technology development and demonstration. Acta Astronautica [online]. 2003, 52(2-6), 317-326 [cit. 2020-06-09]. DOI: 10.1016/S0094-5765(02)00171-6. ISSN 00945765. Dostupné z: <https://www.sciencedirect.com/science/article/abs/pii/S0094576502001716>
- [15] SCHULTHEISS, Daniel. Gravity compensation of deployable solar arrays for small spacecraft. Cambridge, 2003. Diplomová práce. University of Cambridge, Department of Engineering.
- [16] YANGSHENG XU, B. BROWN, S. AOKI a T. KANADE. Mobility And Manipulation Of A Light-weight Space Robot. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems [online]. Raleigh, NC: IEEE, 1992, s. 11-19 [cit. 2020-06-09]. DOI: 10.1109/IROS.1992.587291. ISBN 0-7803-0738-0. Dostupné z: <http://ieeexplore.ieee.org/document/587291/>
- [17] WHITE, G.C. a YANGSHENG XU. An active vertical-direction gravity compensation system. IEEE Transactions on Instrumentation and Measurement [online]. 1994, 43(6), 786-792 [cit. 2020-06-09]. DOI: 10.1109/19.368066. ISSN 00189456. Dostupné z: <http://ieeexplore.ieee.org/document/368066/>
- [18] HAN, Olyvia, David KIENHOLZ, Paul JANZEN a Scott KIDNEY. Gravity-Off-loading System for Large-Displacement Ground Testing of Spacecraft Mechanisms. In: Proceedings of the 40th Aerospace Mechanisms Symposium [online]. Cocoa Beach, FL: NASA, 2010, s. 119-132 [cit. 2020-06-09]. Dostupné z: <https://ntrs.nasa.gov/search.jsp?R=20100021948>
- [19] KIENHOLZ, David A. A Suspension System for Simulating Unconstrained Boundary Conditions. In: Proceedings of the 12th International Modal Analysis Conference. Honolulu: Society for Experimental Mechanics, 1994, s. 1251-1257. ISBN 0912053445.
- [20] SATO, Y., A. EJIRI, Y. IIDA, S. KANDA, T. MARUYAMA, T. UCHIYAMA a H. FUJII. Micro-G emulation system using constant-tension suspension for a space manipulator. In: Proceedings. 1991 IEEE International Conference on Robotics and Automation [online]. Sacramento: IEEE Comput. Soc. Press, 1991, s. 1893-1900 [cit. 2020-06-09]. DOI: 10.1109/ROBOT.1991.131902. ISBN 0-8186-2163-X. Dostupné z: <http://ieeexplore.ieee.org/document/131902/>
- [21] Active Response Gravity Offload System. NASA [online]. Washington, D.C.: NASA [cit. 2020-06-09]. Dostupné z: https://www.nasa.gov/centers/johnson/engineering/integrated_environments/active_response_gravity/
- [22] DUNGAN, Larry K., Paul S. VALLE, Derek R. BANKIERIS, Asher P. LIEBERMAN, Lee REDDEN a Cecil SHY. Active Response Gravity Offload System. USA. US9194977B1. Uděleno 2015.

LITERATURA

- [23] BROWN, H. Benjamin a John M. DOLAN. A Novel Gravity Compensation System for Space Robots. Pittsburgh: Carnegie Mellon University, 1994.
- [24] PAVLÍK, Ondřej. Schéma Zero-G, pasivní. Kuřim: Sobriety s.r.o, 2019.
- [25] MELICHAR, Marek. Mechanický návrh simulátoru dopadu za snížené gravitace [online]. Brno, 2020 [cit. 2020-06-25]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/125112>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce Tomáš Návrát.
- [26] 25.2 Embedded Computers. BISHOP, Robert H. Mechatronic System Control, Logic, and Data Acquisition. Second Edition. Boca Raton, FL: CRC Press, c2008. ISBN 9780849392603.
- [27] 25.1 Introduction. BISHOP, Robert H. Mechatronic System Control, Logic, and Data Acquisition. Second Edition. Boca Raton, FL: CRC Press, c2008. ISBN 9780849392603.
- [28] 4.5 Programmable Logic Controllers. BISHOP, Robert H. The Mechatronics Handbook. Second Edition. Boca Raton, FL: CRC Press, c2002. ISBN 9780849300660.
- [29] 7.1 PLC (PROGRAMMABLE LOGIC CONTROL) CONTROLLERS. ZHANG, Peng. Advanced Industrial Control Technology. Oxford: Elsevier, c2010. ISBN 9781437778076.
- [30] 25.3 Programmable Logic Controllers. BISHOP, Robert H. Mechatronic System Control, Logic, and Data Acquisition. Second Edition. Boca Raton, FL: CRC Press, c2008. ISBN 9780849392603.
- [31] 1.2 REAL-TIME CONTROL SYSTEMS. ZHANG, Peng. Advanced Industrial Control Technology. Oxford: Elsevier, c2010. ISBN 9781437778076.
- [32] 25.3.3 Advanced Capabilities. BISHOP, Robert H. Mechatronic System Control, Logic, and Data Acquisition. Second Edition. Boca Raton, FL: CRC Press, c2008. ISBN 9780849392603.
- [33] 25.4 Conclusion. BISHOP, Robert H. Mechatronic System Control, Logic, and Data Acquisition. Second Edition. Boca Raton, FL: CRC Press, c2008. ISBN 9780849392603.
- [34] 1.1 EMBEDDED CONTROL SYSTEMS. ZHANG, Peng. Advanced Industrial Control Technology. Oxford: Elsevier, c2010. ISBN 9781437778076.
- [35] 1.1.1 Definition and functions. ZHANG, Peng. Advanced Industrial Control Technology. Oxford: Elsevier, c2010. ISBN 9781437778076.
- [36] 25.2.1 Hardware Platforms. BISHOP, Robert H. Mechatronic System Control, Logic, and Data Acquisition. Second Edition. Boca Raton, FL: CRC Press, c2008. ISBN 9780849392603.

- [37] 4 Basic peripherals. HEATH, Steve. Embedded Systems Design. 2nd Edition. Oxford: Newnes, c2002. ISBN 9780750655460.
- [38] 5 Interfacing to the analogue world. HEATH, Steve. Embedded Systems Design. 2nd Edition. Oxford: Newnes, c2002. ISBN 9780750655460.
- [39] 25.2.3 Programming Languages. BISHOP, Robert H. Mechatronic System Control, Logic, and Data Acquisition. Second Edition. Boca Raton, FL: CRC Press, c2008. ISBN 9780849392603.
- [40] 8 Writing software for embedded systems: C extensions. HEATH, Steve. Embedded Systems Design. 2nd Edition. Oxford: Newnes, c2002. ISBN 9780750655460.
- [41] 1.1.2 Architectures and elements. ZHANG, Peng. Advanced Industrial Control Technology. Oxford: Elsevier, c2010. ISBN 9781437778076.
- [42] Preface. TOOLEY, Mike. PC Based Instrumentation and Control. Third edition. Oxford: Butterworth-Heinemann, c2005. ISBN 0750647167.
- [43] National Instruments [online]. Austin, TX: National Instruments, c2020 [cit. 2020-06-12]. Dostupné z: www.ni.com
- [44] 48.2 Digitizing the Analog World. BISHOP, Robert H. The Mechatronics Handbook. Second Edition. Boca Raton, FL: CRC Press, c2002. ISBN 9780849300660.
- [45] 48.1 The Power of Software. BISHOP, Robert H. The Mechatronics Handbook. Second Edition. Boca Raton, FL: CRC Press, c2002. ISBN 9780849300660.
- [46] PC versus PLC, comparing control options. Control Engineering [online]. Downers Grove, IL: Control Engineering, c2020 [cit. 2020-06-12]. Dostupné z: <https://www.controleng.com/articles/inside-machines-pc-versus-plc-comparing-control-options/>
- [47] 9 Industrial computers. ZHANG, Peng. Advanced Industrial Control Technology. Oxford: Elsevier, c2010. ISBN 9781437778076.
- [48] NUCLEO-F767ZI. STMicroelectronics [online]. Geneva: STMicroelectronics, c2020 [cit. 2020-06-12]. Dostupné z: <https://www.st.com/en/evaluation-tools/nucleo-f767zi.html>
- [49] STM32F767ZI. STMicroelectronics [online]. Geneva: STMicroelectronics, c2020 [cit. 2020-06-12]. Dostupné z: <https://www.st.com/en/microcontrollers-microprocessors/stm32f767zi.html>
- [50] UM1905 User Manual: Description of STM32F7 HAL and Low-layer drivers. STMicroelectronics, 2017.
- [51] STM32CubeMX. STMicroelectronics [online]. Geneva: STMicroelectronics [cit. 2020-06-12]. Dostupné z: <https://www.st.com/en/development-tools/stm32cubemx.html>

LITERATURA

- [52] TrueSTUDIO. Atollic [online]. Jönköping: Atollic [cit. 2020-06-12]. Dostupné z: <https://atollic.com/truestudio/>
- [53] DMP 331i. BD SENSORS [online]. BD SENSORS [cit. 2020-06-12]. Dostupné z: <https://www.bdsensors.cz/es/presion/transductores-de-presion/detail/produkt/dmp-i/>
- [54] 37.2 TIA/EIA Serial Interface Standards. BISHOP, Robert H. The Mechatronics Handbook. Second Edition. Boca Raton, FL: CRC Press, c2002. ISBN 9780849300660.
- [55] 14.2 DATA TRANSMISSION I/O DEVICES. ZHANG, Peng. Advanced Industrial Control Technology. Oxford: Elsevier, c2010. ISBN 9781437778076.
- [56] SN65176B, SN75176B DIFFERENTIAL BUS TRANSCEIVERS DATASHEET. Dallas, TX: Texas Instruments, 2003.
- [57] MODBUS over Serial Line Specification and Implementation Guide [online]. V1.02. Hopkinton, MA: Modbus.org, 2006 [cit. 2020-06-12]. Dostupné z: <http://www.modbus.org/specs.php>
- [58] 7 Timers. NORRIS, Donald. Programming with STM32: Getting Started with the Nucleo Board and C/C++. New York: McGraw-Hill Education, 2018. ISBN 1260031314.
- [59] RM0410 Reference manual: STM32F76xxx and STM32F77xxx advanced Arm®-based 32-bit MCUs. STMicroelectronics, 2018.
- [60] JIRKA a spol.: Digitální odměřování [online]. JIRKA [cit. 2020-06-12]. Dostupné z: <https://www.odmerovani.cz/>
- [61] Quadrature Encoder Overview. DYNAPAR [online]. Gurnee, IL: Dynapar, C2020 [cit. 2020-06-12]. Dostupné z: https://www.dynapar.com/technology/encoder_basics/quadrature_encoder/
- [62] 9.4 Digital Filter Design. BISHOP, Robert H. Mechatronic System Control, Logic, and Data Acquisition. Second Edition. Boca Raton, FL: CRC Press, c2008. ISBN 9780849392603.
- [63] Smart sensor: BNO055. BOSCH [online]. BOSCH, c2020 [cit. 2020-06-12]. Dostupné z: <https://www.bosch-sensortec.com/products/smart-sensors/bno055.html>
- [64] Data sheet BNO055: Intelligent 9-axis absolute orientation sensor. Revision 1.6. Reutlingen: Bosch Sensortec, 2019.
- [65] 4 Basic peripherals: I2C bus. HEATH, Steve. Embedded Systems Design. 2nd Edition. Oxford: Newnes, c2002. ISBN 9780750655460.
- [66] 4 Basic peripherals: Serial peripheral interface. HEATH, Steve. Embedded Systems Design. 2nd Edition. Oxford: Newnes, c2002. ISBN 9780750655460.

- [67] MCP3204/3208 datasheet: 2.7V 4-Channel/8-Channel 12-Bit A/D Converters with SPI™ Serial Interface. Microchip Technology Incorporated, 2007.
- [68] IRLZ24N datasheet. El Segundo, CA: International Rectifier, 2002.
- [69] UM1974 User manual: STM32 Nucleo-144 boards. Rev 7. STMicroelectronics, 2017.
- [70] About Qt. Qt Wiki [online]. The Qt Company, 2019 [cit. 2020-06-12]. Dostupné z: https://wiki.qt.io/About_Qt
- [71] All Modules. Qt Documentation [online]. Espoo: The Qt Company, c2020 [cit. 2020-06-12]. Dostupné z: <https://doc.qt.io/qt-5/qtmodules.html>
- [72] Qt Creator. Qt Wiki [online]. The Qt Company, 2016 [cit. 2020-06-12]. Dostupné z: https://wiki.qt.io/Qt_Creator
- [73] QCustomPlot [online]. Bubenreuth: Emanuel Eichhammer, 2018 [cit. 2020-06-12]. Dostupné z: <https://www.qcustomplot.com>

Seznam obrázků

2.1	Příklad pasivního kompenzačního systému [17]	9
2.2	Příklad kompenzačního systému využívajícího pneumatický píst [18]	10
2.3	Aktivního systém kompenzace tíhové síly ARGOS [22]	10
3.1	Schéma systému [24]	13
3.2	Obecná struktura navrhovaného systému [26]	14
5.1	Schéma elektronického subsystému	23
5.2	Snímač tlaku DMP 331i	27
5.3	Rámec protokolu MODBUS [57]	29
5.4	Znak módu RTU [57]	29
5.5	Lineární enkodér TMLS-25	32
5.6	Fázově posunuté signály enkodéru [61]	33
5.7	Snímač BNO055	34
5.8	Snímač U9C	37
5.9	Zapojení solenoidové západky	40
5.10	Schéma napájení elektronického subsystému	43
5.11	Řídicí jednotka	44
6.1	Grafické uživatelské rozhraní	47
6.2	Grafické uživatelské rozhraní, karta ovládání experimentu	48

Seznam zkratk a symbolů

DAC	Digital to Analog Converter
ADC	Analog to Digital Converter
CAN	Controller Area Network
PC	Personal Computer
PLC	Programmable Logic Controller
SFC	Sequential Function Chart
FBD	Function Block Diagram
LD	Ladder-logic Diagram
ST	Structured Text
IL	Instruction List
DMA	Direct Memory Access
FPGA	Field Programmable Gate Array
CPLD	Complex Programmable Logic Device
DSP	Digital Signal Processor
RTOS	Real-Time Operating System
PCI	Peripheral Component Interconnect
USB	Universal Serial Bus
LED	Light-Emitting Diode
SRAM	Static Random-Access Memory
UART	Universal Asynchronous Receiver-Transmitter
USART	Universal Synchronous and Asynchronous Receiver-Transmitter
SPI	Serial Peripheral Interface
CRC	Cyclic Redundancy Check
HAL	Hardware Abstraction Layer
RTU	Remote Terminal Unit
ASCII	American Standard Code for Information Interchange
TTL	Transistor–Transistor Logic

SEZNAM ZKRATEK A SYMBOLŮ

IIR	Infinite Impulse Response
PWM	Pulse-Width Modulation
MOC	Meta-Object Compiler
API	Application programming interface

Seznam příloh na CD

Příloha 1 - zdrojový kód programu pro STM32F767ZI

Příloha 2 - zdrojový kód grafického uživatelského rozhraní